



RESEARCH CENTER
Bordeaux - Sud-Ouest

FIELD

Activity Report 2013

Section Software

Edition: 2014-03-19

1. ALEA Project-Team	4
2. BACCHUS Team	5
3. CAGIRE Team	9
4. CARMEN Team	11
5. CEPAGE Project-Team	12
6. CONCHA Project-Team	14
7. CQFD Project-Team	20
8. FLOWERS Project-Team	21
9. GEOSTAT Project-Team	44
10. HIEPACS Project-Team	45
11. LFANT Project-Team	50
12. MAGIQUE-3D Project-Team	53
13. MAGNOME Project-Team	55
14. MANAO Team	57
15. MC2 Project-Team	59
16. MNEMOSYNE Team	62
17. PHOENIX Project-Team	64
18. POTIOC Team	71
19. REALOPT Project-Team	72
20. RUNTIME Project-Team	73

ALEA Project-Team

5. Software and Platforms

5.1. BiiPS software

BiiPS is a general software, developed by Adrien Todeschini, for Bayesian inference with interacting particle systems, a.k.a. sequential Monte Carlo (SMC) methods. It aims at popularizing the use of these methods to non-statistician researchers and students, thanks to its automated “black box” inference engine.

It borrows from the BUGS/JAGS software, widely used in Bayesian statistics, the statistical modeling with graphical models and the language associated with their descriptions.

Unlike MCMC methods used by BUGS/JAGS, SMC methods are more adapted to dynamic problems (tracking, signal filtering, etc).

A beta version of the software can be downloaded from the website of the [BiiPS project](#). This software has been presented at the international workshop [BayesComp](#) in Kyoto, the international conference [ISBA](#) in Tokyo, the conference on [Premières Rencontres R](#) in Bordeaux, and the [international workshop on efficient simulation in finance](#) in Paris.

Adrien Todeschini participated to a session focused on Recent Developments in Software for MCMC (and SMC) (MCMSki IV, Fifth IMS-ISBA joint meeting MCMSki IV, Chamonix Mont-Blanc, France).

This invited panel features four leading researchers working on software development for Bayesian computation. Each panelist will highlight their particular software, including its history, development, and relative strengths and weaknesses. Looking forward, panelists will discuss and debate the future of Bayesian computation and software development, including challenges, opportunities and bottlenecks. Emphasis throughout will be on simplifying and automating the implementation of Monte Carlo methods, with an eye towards scalability to larger and more complex models and data.

BACCHUS Team

5. Software and Platforms

5.1. AeroSol

Participants: Dragan Amenga-Mbengoué [Bacchus], Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Francois Pellegrini [Bacchus], Vincent Perrier [Corresponding member], Francois Rué [Bacchus], Mario Ricchiuto [Bacchus].

The AeroSol software is jointly developed in teams Bacchus and Cagire. It is a high order finite element library written in C++. The code has been designed so as to allow for efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPA are used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances. The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

This year some important features were added including : definition of CMake options for optimization and for using different compilers (GNU gcc, Intel icc, and IBM xlc) ; new element classes (lagrange and hierarchical orthogonal finite element basis for pyramids, Gauss Lagrange elements) ; implicit time integrators (backward Euler, Crank-Nicolson, and BDF from 2nd to 6th order) ; anisotropic diffusion models and (compressible) Navier-Stokes models ; debuggin by looging at memory traces with an interfacing with the PAPI library (tests have also been performed with VTUNE and TAU) ; improvements in schemes robustness and efficiency (Galerkin discretization of advection optimized by stocking most of the geometrical functions and finite elements computations, explicit and implicit version of the DG discretization of diffusion problems, implementation of Taylor-Galerkin stabilization and simplified SUPG stabilization) ; boundary conditions (time dependent, periodic, non reflecting) ; low Mach numerical fluxes for DG ; development of steady and unsteady tests related to all these new features.

5.2. COCA

Participants: Mario Ricchiuto [corresponding member], Gérard Vignoles, Gregory Perrot.

COCA(CodeOxydationCompositesAutocicatrissants) is a Fortran 90 code for the simulation of the oxidation process in self-healing composites COCA solves the discrete finite element equations relative to the oxidation (chemistry) and flow (potential) models. Time integration is performed with an implicit approach (Backward Euler or second order backward differencing). The linear algebraic systems arising in the discretization are solved with the MUMPSlibrary.

5.3. RealfluidS

Participants: Dante de Santis, Gianluca Geraci, Pietro Marco Congedo, Rémi Abgrall [corresponding member].

RealfluidS is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multimaterial, MHD flows and turbulent flows (using the SA model). There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. This software implements the more recent residual distribution schemes. The code has been parallelized with and without overlap of the domains. The uncertainty quantification library RobUQhas been coupled to the software. A partitioning tool exists in the package, which uses Scotch. Recently, the code has been developed for taking into account real-gas effects, in order to use a whatever complex equation of state. Further developments concerning multiphase effects are under way.

5.4. MMG3D

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

MMG3D is a tetrahedral fully automatic remesher. Starting from a tetrahedral mesh, it produces quasi-uniform meshes with respect to a metric tensor field. This tensor prescribes a length and a direction for the edges, so that the resulting meshes will be anisotropic. The software is based on local mesh modifications and an anisotropic version of Delaunay kernel is implemented to insert vertices in the mesh. Moreover, MMG3D allows one to deal with rigid body motion and moving meshes. When a displacement is prescribed on a part of the boundary, a final mesh is generated such that the surface points will be moved according this displacement. MMG3D is used in particular in GAMMA for their mesh adaptation developments, but also at EPFL (maths department), Dassault Aviation, Lemma (a french SME), etc. MMG3D can be used in FreeFem++ (<http://www.freefem.org>), a free software which eases the solving of PDEs and in Gmsh (<http://geuz.org/gmsh/>). More details can be found on <http://www.math.u-bordeaux1.fr/~doj/logiciels/mmg3d.php>.

A new version of MMG3D is under development. The big novelty of this version is the modification of the surface triangulation. A. Froehly, ingénieur in the FUI Rodin, is working on this new version.

5.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Dante de Santis, Maria-Giovanna Rodio.

The ORComp platform is a simulation tool permitting to design an ORC cycle. Starting from the solar radiation, this platform computes the cycle providing the best performance with optimal choices of the fluid and the operating conditions. It includes RobUQ, a simulation block of the ORC cycles, the RealfluidScore for the simulation of the turbine and of the heat exchanger, the software FluidProp (developed at the University of Delft) for computing the fluid thermodynamic properties.

5.6. PaMPA

Participants: Cédric Lachat, François Pellegrini [Corresponding member], Cécile Dobrzynski, Hervé Guillard [PUMAS], Laurent Hascoët [Tropics].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of *entities* (elements, faces, edges, nodes, etc.), linked by *relations* (that is, computation dependencies).

PaMPA interfaces with Scotch for mesh redistribution, and with MMG3D for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged-in, in order to handle other types of elements.

Following the PhD defense of Cédric Lachat last December, version 1.0 is about to be released publicly under the GPL license. This version allows users to declare distributed meshes, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh in parallel the areas of a mesh that need to be emeshed, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture.

PaMPA is already used as the data structure manager for two solvers being developed at Inria: Plato (team PUMAS) and AeroSol (teams BACCHUS and CAGIRE).

5.7. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Gianluca Geraci, Maria Giovanna Rodio, Kunkun Tang, Julie Tryoen.

The RobUQ platform has been conceived to solve problems in uncertainty quantification and robust design. It includes the optimization code ALGEN, and the uncertainty quantification code NISP. It includes also some methods for the computation of high-order statistics, efficient strategies for robust optimization, the Simplex2 method. Some methods are developed in partnership with the Stanford University (in the framework of the associated team AQUARIUS). Other methods are developed in the context of ANR UFO.

5.8. Scotch

Participants: François Pellegrini [corresponding member], Sébastien Fourestier.

parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, mesh partitioning.

Scotch (<http://www.labri.fr/~pelegri/scotch/>) is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of Scotch was to compute high-quality static mappings of valued graphs representing parallel computations onto target architectures of arbitrary topologies. This allows the mapper to take into account the topology and heterogeneity of the target architecture in terms of processor speed and link bandwidth. This feature, which was meant for the NUMA machines of the 1980's, has not been widely used in the past because machines in the 1990's became UMA again thanks to hardware advances. Now, architectures become NUMA again, and these features are regaining popularity.

The Scotch package consists of two libraries: the sequential Scotch library, and the parallel PT-Scotch library (for "*Parallel Threaded Scotch*") that operates according to the distributed memory paradigm, using MPI. Scotch was the first full 64-bit implementation of a general purpose graph partitioner.

Version 6.0, released on December 2012, corresponding to the 20th anniversary of Scotch, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential Scotch library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be available to the parallel PT-Scotch library in the upcoming release 6.1.

Scotch has been integrated in numerous third-party software, which indirectly contribute to its diffusion. It is natively available in several Linux and Unix distributions, as well as on some vendors platforms (SGI, etc).

5.9. SLOWS

Participants: Mario Ricchiuto [corresponding member], Andrea Filippini.

SLOWS ("*Shallow-water fLOWS*") is a C-platform allowing the simulation of free surface shallow water flows with friction. Arbitrary bathymetries are allowed, defined either by some complex piecewise analytical expression, or by *xyz* data files, the classical Manning model for friction is used, and an Exner model is implemented for sediment transport. For non-hydrostatic propagation the enhanced Boussinesq equations of Madsen and Sorensen are used. The equations are discretized with a residual based approach which is an adaptation of the schemes developed for aeronautics applications. Due to the inherent unsteadiness of these flows, the time discretization plays an important role. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach. Newton and frozen Newton loops are used to solve the implicit nonlinear equations. The linear algebraic systems arising in the discretization are solved with the MUMPSlibrary.

5.10. Nomesh

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

Nomesh is a software allowing the generation of third order curved simplicial meshes. Starting from a "classical" mesh with straight elements composed by triangles and/or tetrahedra, we are able to curve the boundary mesh. Starting from a mesh with some curved elements, we can verify if the mesh is valid, that means there is no crossing elements and only positive Jacobian. If the curved mesh is non valid, we modify it using linear elasticity equations until having a valid curved mesh.

CAGIRE Team

5. Software and Platforms

5.1. AeroSol

Participants: Dragan Amenga-Mbengoué [Bacchus], Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Francois Pellegrini [Bacchus], Vincent Perrier [Cagire, correspondant], Francois Rué [Bacchus], Mario Ricchiuto [Bacchus].

The software AeroSol is jointly developed in the team Bacchus and the team Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes.

The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPA are used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances.

The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

This year, Simon Delmas and Yann Moguen were recruited within the team Cagire. Their respective development, low Mach solver for compressible flows and turbulence injection boundary conditions are performed in the library Aerosol. At the end of 2012, Aerosol had the following features

- **development environment** use of CMake for compilation, CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests.
- **In/Out** link with the XML library for handling with parameter files. Reader for GMSH, and writer on the VTK-ASCII legacy format (cell and point centered). Parallel GMSH reader, XML paraview files on unstructured meshes (vtu) and parallel XML based files (pvtu).
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms and hexaedra. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra.
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra and hexaedra.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping.
- **Linear Solvers** link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes.

- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Discontinuous Galerkin methods for hyperbolic systems. SUPG and Residual distribution schemes.
- **Numerical fluxes** centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. collective asynchronous communications (PaMPA based). Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM.
- **C++/Fortran interface** Tests for binding fortran with C++.

This year, the following features were added

- **development environment** Definition of CMake options for optimization and for using different compilers. Currently, the following compilers have been tested: GNU gcc, Intel icc, and IBM xlc. Aerosol can now be linked with HDF5, PAPI, and can use different BLAS implementations like eigen or MKL.
- **In/Out** Point centered visualization for discontinuous approximations. XML binary output for Paraview was added. The link with HDF5 was added for parallel IO for defining XDMF format. A geometrical pre-partitioning was developed for reducing the size of the parallel graph in the parallel mesh reading.
- **Pyramids** Mesh reader, Lagrange and hierarchical orthogonal finite element basis were added for pyramids. Geometrical functions for linear pyramids were also added.
- **Finite element** Gauss Lagrange finite element basis (order 1 and 2) for triangles.
- **Time iteration** the following implicit integration schemes were added: backward Euler, Crank-Nicolson, and BDF from 2nd to 6th order.
- **Linear Solvers** Interface with PETSc was tested on a parallel environment. An in house block diagonal solver was developed.
- **Memory handling** Aerosol can now work on hybrid meshes.
- **Models** the generic model interface supports now diffusive models. Anisotropic diffusion and (compressible) Navier-Stokes models were added.
- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU.
- **Parallel computing** Tests were performed on the clusters Pyrene (Université de Pau), poincaré (Maison de la Simulation), and on the Tier-1 cluster Turing (IDRIS).
- **Numerical schemes** The DG discretization of advection problems was optimized by stocking most of the geometrical functions and finite elements computations, and by using BLAS implementations for linear computations. Implicit versions of the DG discretization of advection problems. Development of explicit and implicit version of the DG discretization of diffusion problems. Time dependent boundary conditions, periodic boundary conditions, non reflecting boundary conditions. Development of low Mach numerical fluxes, and development of stationary and unstationary tests for this kind of problem.

CARMEN Team

5. Software and Platforms

5.1. CEPS: a Cardiac ElectroPhysiology Simulator

The Carmen team develops a software code to perform high performance numerical simulations in cardiac electrophysiology using unstructured three-dimensional grids. The software, called CEPS (*Cardiac Electro-physiology Simulation*), is developed as a common tool for researchers in the Carmen team and for our partners and colleagues in scientific computing and biomedical engineering. The goal of CEPS is to easily allow the development of new numerical methods and new physical models. Thanks to an ADT, actual developments started at the end of 2012 and still continue.

As compared to other existing softwares, CEPS aims at providing a more general framework of integration for new methods or models and a better efficiency in parallel. CEPS is designed to run on massively parallel architectures, and to make use of state-of-the-art and well known computing libraries to achieve realistic and complex heart simulations. CEPS also includes software engineering and validation tools [30]. We use the platform GForge (gforge.inria.fr/projects/ceps) based on Subversion. This allows to keep a history of developments for developers and users.

Some of our collaborators actively participate to the testing and discussion for the development of CEPS, namely:

- C. Pierre, LMA Université de Pau et des Pays de l'Adour;
- R. Turpault, LMA Université de Nantes;
- L. Gerardo-Giorda, BCAM Bilbao.

5.2. PROPAG

The workhorse for our applied simulation studies of the whole human heart is PROPAG, a code that has its origins at the Université de Montréal in Canada, and has been further developed by the Institute of Computational Science in Lugano, Switzerland. PROPAG is highly configurable, runs with complex model geometries, and runs efficiently on high-performance computing systems with many thousands of cores. It is particularly useful for whole-heart studies, which typically rely on very large model sizes (in the order of 10^8 elements), several different membrane models and cell types in a single simulation run, and several regionally varying parameters.

PROPAG is presently used in our group to study the relation between the substrate, complexity, and electrocardiographic features of atrial fibrillation and of cardiomyopathy-related ventricular arrhythmia, providing the efficiency and flexibility that is required to handle the complex anatomical structures that are involved.

5.3. Model construction – A new project

Many of our projects rely on realistic or even patient-tailored meshes to represent the anatomy of the human heart and torso. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. We presently use a variety of in-house and public software packages to perform this work and are able to produce meshes of sufficient quality, but we strive for an important streamlining of this work. We have initiated a discussion with several groups inside and outside Inria who have similar needs or can offer solutions. We specifically investigate the possibility to build a common software which combines and complements our present solutions. The new code should make various methods easily accessible and automate the work as much as possible. Because accuracy and mesh quality are important requirements, the new code should also provide convenient options for human intervention where algorithms fall short. For example, manual segmentation and mesh editing should be as easy and efficient as they are in medical-imaging tools and 3D-editing software, respectively, but well integrated into the workflow.

CEPAGE Project-Team

5. Software and Platforms

5.1. SimGrid

Participants: Paul Renaud-Goud, Lionel Eyraud-Dubois [correspondant].

SimGrid (<http://simgrid.gforge.inria.fr/>) is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. It is based on experimentally validated models, and features very high scalability, which allows to perform very large scale simulations. It is used by over a hundred academic users all over the world, and has been used in about one hundred scientific articles.

CEPAGE has contributed to this software by participating in the management of the project and in many design decisions. As a part of the SONGS project, we also participated in the development and validation of new models and interfaces for the HPC and Cloud Computing platforms.

Software assessment : A-4, SO-4, SM-4, EM-3, SDL-5

Contribution : DA-2, CD-2, MS-2, TPM-3.

5.2. Hubble

Participants: Ludovic Courtes, Nicolas Bonichon [correspondant].

Hubble is implemented in Scheme, using GNU Guile version 2. Details of the simulation, such as keeping track of processor occupation and network usage, are taken care of by SimGrid, a toolkit for the simulation of distributed applications in heterogeneous distributed environments.

The input to Hubble is an XML description of the DAG of build tasks. For each task, a build duration and the size in bytes of the build output are specified. For our evaluation purposes, we collected this data on a production system, the <http://hydra.nixos.org/> build farm hosted at the Technical University of Delft. The DAG itself is the snapshot of the Nix Package Collection (Nixpkgs) corresponding to this data. Hubble has its own in-memory representation of the DAG in the form of a purely functional data structure.

The Nixpkgs DAG contains fixed-output nodes, i.e., nodes whose output is known in advance and does not require any computation. These nodes are typically downloads of source code from external web sites. The raw data collected on <http://hydra.nixos.org/> specifies a non-zero duration for these nodes, which represents the time it took to perform the download. This duration info is irrelevant in our context, since they don't require any computation, and Hubble views these nodes as instantaneous.

See also the web page <http://hubble.gforge.inria.fr/>.

Software assessment: A-3, SO-3, SM-2, EM-1, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

5.3. Gengraph

Participant: Cyril Gavaille [correspondant].

This is a command-line tool for generating graphs. There are several output formats, includes the dot format from GraphViz. It generates also .pdf files for visualization. Several graph algorithms have been implemented (diameter, connectivity, treewidth, etc.) which can be tested on the graphs. The software has been originally designed for teaching purpose so that students can test their project algorithms on many non trivial families like random geometric graphs, graphs of given density, given treewidth. It is also used for research purpose, in particular the exhaustive search results in the Emilie Diot's thesis are based on `gengraph`. The program can filter a list of graphs based to many criteria, as for instance it can extract all graphs of a given list that are 2-connected, of diameter at least four, and that exclude some minor (or some induced subgraph).

Currently, more than 100 parametrized graph families are implemented, supporting simple operators like complementation, random edge/vertex removal, and others. The source has more than 10,000 lines including a command-line documentation of 2,000 lines. The single source file is available at <http://dept-info.labri.fr/~gavoille/gengraph.c>

Software assessment: A-3, SO-3, SM-2, EM-2, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

5.4. Bedibe

Participants: Lionel Eyraud-Dubois [correspondant], Przemyslaw Uznanski.

Bedibe (Benchmarking Distributed Bandwidth Estimation) is a software to compare different models for bandwidth estimation on the Internet, and their associated instantiation algorithms. The goal is to ease the development of new models and algorithms, and the comparison with existing solutions. Additionally, we have developed a measuring framework which can be deployed on PlanetLab to obtain available bandwidth datasets.

See also the web page <http://bedibe.gforge.inria.fr/>.

Software assessment : A-1-up2, SO-3, SM-1-up2, EM-2, SDL-1-up2.

5.5. MineWithRounds

Participants: Sofian Maabout [correspondant], Nicolas Hanusse.

The software implements a parallel algorithm aiming at computing *Borders* that's sets of maximal/minimal subsets of objects satisfying some anti-monotone condition. It is implemented in C++ together with the openMP library to exploit multi-core machines. In its current status, it outperforms state of the art implementations addressing the Maximal Frequent Itemsets problem.

Software assessment: A-2, SO-4, SM-2, EM-2, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

5.6. FSM: Full Skycube Materialization

Participant: Sofian Maabout [correspondant].

The software implements a parallel algorithm aiming at fully materializing skycubes: the set of all possible skyline queries. It is implemented in C++ together with the openMP library to exploit multi-core machines. In its current status, it outperforms state of the art implementations such as QSkycube and PSkyCube (VLDB Journal '13)

Software assessment: A-2, SO-4, SM-2, EM-2, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

ADT: Pierre Matri has been hired as an engineer for two year starting from September 15th, 2013. He is in charge to implement our algorithm on top of a large scale infrastructure especially by using Map-Reduce paradigm and its Hadoop implementation. His main focus during 2013 concerned the functional dependencies extraction from large distributed data bases by considering exact and approximate solutions.

CONCHA Project-Team

5. Software and Platforms

5.1. C++ library Concha

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

The objectives of our library CONCHA are to offer a flexible and extensible software with respect to:

- Numerical methods and
- Physical models.

The aim is to have a flexible code which could easily switch between the different discretizations, in order to provide a toolbox for rapid testing of new ideas.

The software architecture is designed in such a way that a group of core developers can contribute in an efficient manner, and that independent development of different physical applications is possible. Further, in order to accelerate the integration of new members and in order to provide a basis for our educational purposes (see Section 9.2), the software proposes different entrance levels. The basic structure consists of a common block, and several special libraries which correspond to the different fields of applications described in Sections – Hyperbolic solvers, Low-Mach number flow solvers, DNS, and viscoelastic flows. A more detailed description of each special library may be found below. In order to coordinate the cooperative development of the library, Concha is based on the Inria-Gforge.

5.2. User interface and python interface

Participants: Roland Becker, David Trujillo.



Figure 6. Graphical user interface: option panel (left) and process panel (right) of the install tool.

We are confronted with heterogenous backgrounds and levels of implication of the developers and users. It seems therefore crucial to be able to respond to the different needs. Our aim is to facilitate the development of the library, and at the same time, to make it possible that our colleagues involved in physical modeling can have access to the functionality of the software with a reasonable investment of time. Two graphical user interfaces have been developed: one for the installation of the library and another one for the building and execution of projects. They are based on common database and scripts written in python. The scripts can also be launched in a shell. In Figure 6 the user interface of the install tool is shown. The option panel allows to choose the components for conditional compilation and the compilation type (debug and release).



Figure 7. Graphical user interface: project build panel (left) and parameter panel (right) of the project tool.

In Figure 7 the user interface of the project tool is shown. A project consists of a number of sources files and a parameter file used by the C++-executable. The sources define classes derived from the library, which are used to specify certain data such as boundary conditions and employed finite element spaces. The parameter file contains algorithmic information and physical parameters. It is generated from a database by the python utilities.

The tools offered by this development platform are based on a python interface for the library, called pyConcha. It offers a common interface, based on a pluggin-system, which allows the developpement of command line tools in parallel. This year the consolidation of the interface part of pyConcha has been an important task. The pyConcha library is now a framework rather than a simple interface to Concha C++ library. It allows now creation of plugins, so that each user-programmer can customize pyConcha to his own goals. Previously, two main programs were working: concha-install.py to install library, and concha-project.py for (semi-)end-users. Both are now plugins of pyConcha, and can be launched by pyConcha at startup. A plugin visualization could now be developed in an independant way, and launched by pyConcha on demand.

The structure of pyConcha framework is clearly splitted in various modules(layers): Command Line Interface module, Graphical User Interface module and Handlers modules, see Figure 8 . A great effort has been made for internationalization of pyConcha.

5.3. Euler equations

Participants: Roland Becker, Kossivi Gokpi, Robert Luce, Eric Schall, David Trujillo.



Figure 8. Structure of the pyConcha framework.

Based on the library CONCHA we have developed a solver for hyperbolic PDE's based on DGFEM. So far different standard solvers for the Euler equations such as Lax-Friedrichs, Steger-Warming, and HLL have been implemented for test problems. A typical example is the scram jet test case shown in Figure 9 .



Figure 9. Computed Mach-number distribution for the Scramjet test problem.

5.4. Incompressible flow solvers

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

We have started the validation of the implementation of different finite element methods for incompressible flows at hand of standard benchmark problems as the Stokes flow around a symmetric cylinder [58] and the stationary flow around a slightly non symmetric cylinder [63], see Figure 10 .

5.5. DNS

Participants: Roland Becker, David Trujillo.

```
../../../../projets/concha/IMG/hulsen.png
```

```
../../../../projets/concha/IMG/cylinderBenchmark.png
```

For the direct numerical simulation of incompressible turbulent flows, we have started to develop a special solver based on structured meshes with a fast multigrid algorithm incorporating projection-like schemes. The main idea is to use non-conforming finite elements for the velocities with piecewise constant pressures, leading to a special structure of the discrete Schur complement, when an explicit treatment of the convection and diffusion term is used.

5.6. Validation and comparison with other CFD-software

Participants: Roland Becker, Didier Graebbling, Eric Schall, David Trujillo.

Validation and comparison with other CFD-software is crucial in order to evaluate the potential of our numerical schemes concerning accuracy, computing time and other practical aspects.

We have compared the Concha library for incompressible and compressible flows. For incompressible flows, we have used a test case proposed by Hulsén and the well-known Schäfer-Turek cylinder benchmark in order to validate the accuracy of the Stokes and Navier-Stokes solvers. The viscoelastic code has been compared with PolyFlow for different test configurations.

The compressible Euler code has been compared to the ELSA software developed by ONERA.

For further comparison and validation, it would be important to consider other commercial and research tools such as: *Aéro3* (Inria-Smash), AVBP (CERFACS), Fluent (ANSYS), and OpenFOAM (OpenCfd).

For this purpose we have proposed the ADT-project VALSE in collaboration with a small company involved in aerodynamics (EPSILON Toulouse), which unfortunately has been rejected by Inria.

CQFD Project-Team

5. Software and Platforms

5.1. Package `edrGraphicalTools`

This R package gives graphical tools for selecting the number of slices and the dimension of the model in SIR and SAVE approaches. It also provides the estimation of the reduction dimension subspace and the non parametric estimation of the link function using smoothing techniques. The package is available via the link <http://cran.r-project.org/web/packages/edrGraphicalTools/index.html>.

5.2. Package `ClustOfVar`

This R package is dedicated to cluster analysis of a set of variables. Variables can be quantitative, qualitative or a mixture of both. A new version 0.8 of the package is available since december 2013 via the link <http://cran.r-project.org/web/packages/ClustOfVar/index.html>. This version provides now a function to predict values of new observations on the synthetic variables of the clusters. This new function was used for supervised classification and variable selection in gene expressions data [42].

5.3. Package `PCAmixdata`

This package is dedicated to factorial analysis and rotation of quantitative data, qualitative data, or mixed data. The PCAMIX method, proposed in this package includes the ordinary principal component analysis (PCA) and multiple correspondence analysis (MCA) as special cases. Orthogonal varimax rotation of the principal components of PCAMIX is also implemented in this package. This year, a new method has been developed for Multiple Factorial Analysis in case of mixtures of quantitative and qualitative variables within groups. It was implemented in the package and presented to the 2èmes Rencontres R in Lyon [44], and to the 45èmes Journées de Statistique in Toulouse [41].

FLOWERS Project-Team

5. Software and Platforms

5.1. Perception Tools

Participants: David Filliat [correspondant], Natalia Lyubova, Louis-Charles Caron, Alexander Gepperth.

5.1.1. Perception Abstraction Engine

Participants: David Filliat [correspondant], Natalia Lyubova.

PAE (Perception Abstraction Engine) is a C++ library developed to provide a uniform interface to existing visual feature detector such as SIFT, SURF, MSER, superpixels, etc... Its main goal is to be able to use these various feature detectors in a "bag of feature" approach for applications such as robot localisation and object recognition. Several approach are also implemented for the visual vocabularies, in particular the fast incremental vocabularies developed in the team.

The library provide common C++ interfaces to feature detectors, visual features and visual vocabularies. A factory approach make it possible to change the feature detectors and visual vocabularies types and parameters through configuration strings, without the need to recompile. Some applications are also included in the library, in particular topological robot localization (room recognition) and visual object recognition. An Urbi interface is also provided for these modules.

5.1.2. Incremental object discovery

Participants: Natalia Lyubova [correspondant], David Filliat.

This software makes it possible to detect, model and recognize objects in a scenario of interaction between a humanoid robot and a human teacher. It is based either on standard images, or on the kinect camera to take advantage of the depth information. The software is written in C++ and relies mainly on PAE and OpenCV.

The software implements several modules: candidate object segmentation based on motion information, keypoint-based object tracking, incremental object model construction integrating multiple features (keypoints + superpixels) and object categorisation based on mutual information with robot motors (making it possible to segment robot parts, objects and humans). Based on all these modules, it is possible for the robot to learn objects shown by a human partner and to improve the objects models by manipulating them when they are put in front of the robot.

5.1.3. Object recognition from a 3-D point cloud

Participants: Louis-Charles Caron [correspondant], Alexander Gepperth, David Filliat.

This software scans the 3-D point cloud of a scene to find objects and match them against a database of known objects. The process consists in 3 stages. The segmentation step finds the objects in the point cloud, the feature extraction computes discriminating properties to be used in the classification stage for object recognition.

The segmentation is based on simple assumptions about the geometry of an indoor scene. Successive RANSACs are used to find large planes, which correspond to the floor, ceiling and walls. The cloud is stripped from the points belonging to these planes. The remaining points are clustered, meaning that close-by points are considered to form a single object.

Objects are characterized by their shape and color. Color histograms and SIFT features are computed, using the PAE library, to capture the visual appearance of the objects. Their shape is encoded by computing thousands of randomly chosen SURFLET features to construct a relative frequency histogram.

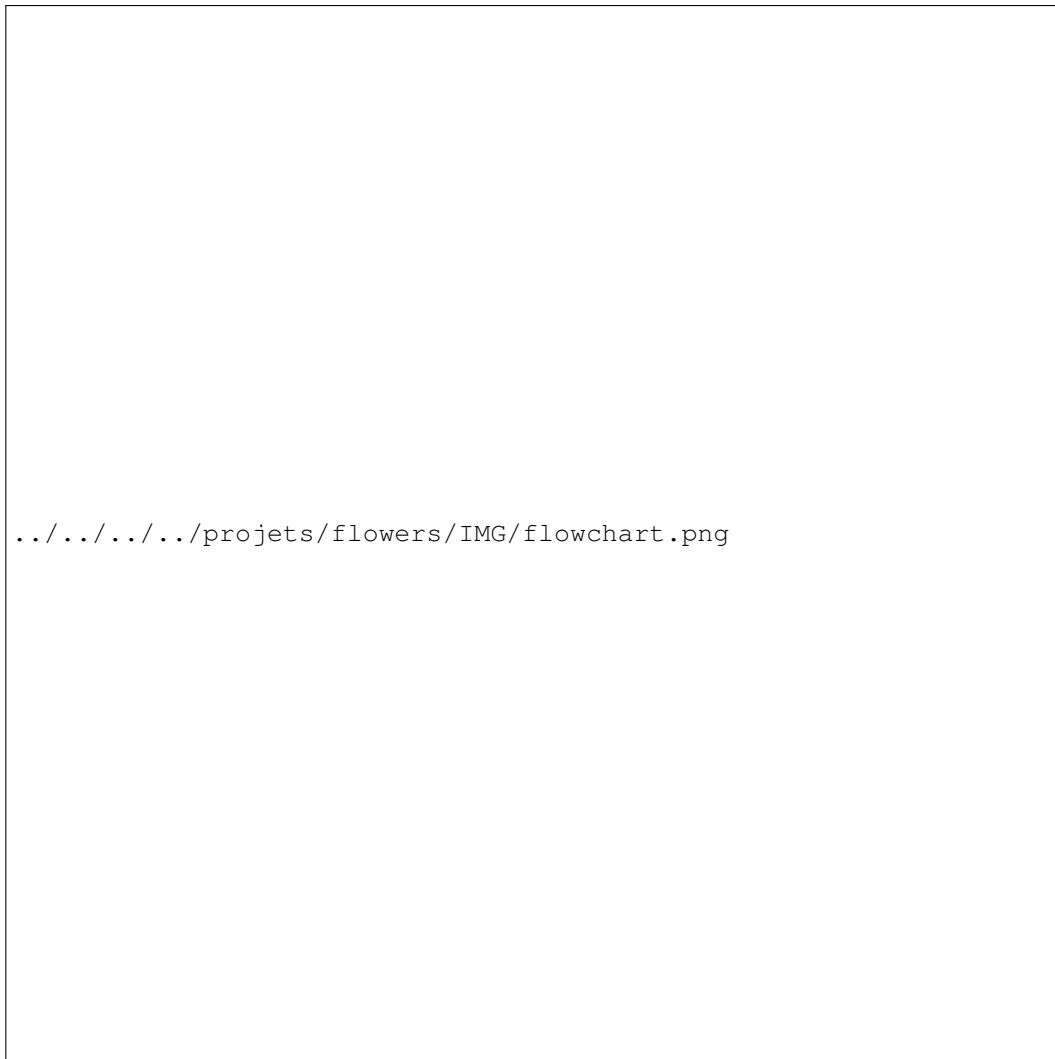


Figure 1. System Overview of the Incremental object discovery Software.

An early classification is done using each of the 3 features separately. For the color features a bag of words approach (from PAE) is used. For the shape feature, the minimum squared distance between the object's histogram and that of all objects in the database is calculated. Classification scores are then fused by a feed-forward neural network to get the final result [81].

5.1.4. PEDDETECT: GPU-accelerated person detection demo

Participant: Alexander Gepperth [correspondant].

PEDDETECT implements real-time person detection in indoor or outdoor environments. It can grab image data directly from one or several USB cameras, as well as from pre-recorded video streams. It detects multiple persons in 800x600 color images at frame rates of >15Hz, depending on available GPU power. In addition, it also classifies the pose of detected persons in one of the four categories "seen from the front", "seen from the back", "facing left" and "facing right". The software makes use of advanced feature computation and nonlinear SVM techniques which are accelerated using the CUDA interface to GPU programming to achieve high frame rates. It was developed in the context of an ongoing collaboration with Honda Research Institute USA, Inc.

5.1.5. A Python OptiTrack client

Participant: Pierre Rouanet [correspondant].

This python library allows you to connect to an OptiTrack from NaturalPoint (<http://www.naturalpoint.com/optitrack/>). This camera permits the tracking of 3D markers efficiently and robustly. With this library, you can connect to the Motive software used by the OptiTrack and retrieve the 3D position and orientation of all your tracked markers directly from python.

5.2. Datasets

5.2.1. Choreography dataset 1 and 2

Participants: Olivier Mangin [correspondant], Haylee Fogg.

These databases contain choreography motions recorded through a kinect device. In the first dataset, these motions have a combinatorial structure: from a given set of primitive dance motions, choreographies are constructed as simultaneous execution of some of these primitive motions. Primitive dance motions are chosen from a total set of 48 motions and are spanned over one or two limbs, either the legs (e.g. walk, squat), left or right arm (e.g. wave hand, punch) or both arms (e.g. clap in hands, paddle). Complex choreographies are produced as the simultaneous demonstration of two or three of these primitive motion: either one for legs and one for both arm, or one for legs and one for each arm. The dataset has been used in the experiments from [104] for studying learning techniques allowing to identify dictionaries of motion primitives, and is publicly available at https://flowers.inria.fr/choreography_database.html.

The second dataset only contains choreographies composed of a single motion. It contains 110 records of each gesture from a set of 10 simple gestures and was used in the experiments from [53]. The dataset is publicly available at <https://flowers.inria.fr/choreo2>.

5.2.2. Development on NoFish Platform

Participants: Paul Fudal [correspondant], Sao Mai Nguyen.

NoFish (fig. 2) platform is a setup used by Mai Nguyen to perform several experiments following her PhD work on social learning and intrinsic motivation. The setup consists to an ErgoRobot (fig. 3) with a fishing rod attached to the tip and a cap made with a red juggling ball. The robot is plugged on an ethernet power-switch used to turn it off if something wrong happens with the robot (for example : the robot try to go in a position he cannot reach and its motors are forcing too much). Tracking the cap is made with Full HD camera on the ceiling. At last, a video-projector prints informations on the floor (fig. 4) which helps for interactions between humans and the robots.



Figure 2. Illustration of the NoFishPlatform



Figure 3. Illustration of NoFish ErgoRobot



Figure 4. NoFish cap and informations printed on the floor

Controlling the robot is made using URBI¹ scripts allowing different control level from single motor control to pre-programmed primitive (for example : resetting the robot or make it going to its starting position). URBI is also used to perform action through the power-switch if the robot must be turned off.

Tracking the cap is made by a program written using OpenCV and keep up to date the cap's coordinates by sending them to URBI through a network socket.

An other program written in JAVA and Processing² allows to print informations on the floor; it consists of a server receiving through a network socket texts and shapes informations to print on the floor and a JAVA object which can be used with compatible software. This component were really useful for giving simple and direct information during an experiment and also during interaction between a human and the robot for socially guided experiments.

The main program is written in Matlab and includes all the explained previous components and softwares. It also includes a forward kinematic calculation module used to ensure a movement sent can be safely be played by the robot to avoid it breaking himself; this module gives step by step informations of what will happen when the robot will play the movement like, for example, if the fishing cap will touch the floor at the end or not which permits to keep the robot safe and speed up experiments by ignoring dangerous moves for the robot and useless ones for the deployed algorithm.

This setup were used by Sao Mai Nguyen to run experiments during her thesis on social learning and intrinsic motivation

5.3. Learning algorithms

5.3.1. KidLearn

Participants: Manuel Lopes [correspondant], Benjamin Clement, Pierre-Yves Oudeyer, Didier Roy.

The KidLearn software provides an Intelligent Tutoring System that optimizes teaching sequences based on the estimated level of each particular student [65]. We implemented a Game of Money that allows students, ages 7-8, to learn how to use money. It includes 3 main components: i) a webserver that handles the requests and stores the experiments in a database; ii) a GUI that provides the interface for the game; and iii) the optimization software.

Graphical interfaces in ITS can have unwanted side effects. For this reason, the interface was entirely designed with the help of a didactician, with several specific design choices motivated by pedagogical, motivational and attention requirements. For example, the interface, shown in Figure 5, is such that:

- display is as clear and simple as possible;
- there is no chronometer, so that students are not put under time pressure;
- coins and banknotes have realistic visual appearance, and their relative sizes are respected;
- display of prices use visual encodings commonly used in shops;
- the zone for receiving money is automatically cleared in case of error after the student submits it;
- automatic snapping of money and tokens icons in the reception zone, and automatic visual arrangement;
- text quantity is kept to minimum;

Four principal regions are defined in the graphical interface, as shown in Figure 5. The first is the wallet location where users can pick and drag the money to drop them on the repository location to make for the correct price. The object and the price are present in the object location, where the price can be written and/or spoken depending on the parameterization of the activity. The information location is used to display information for the learners such as extra clues when they make a mistake (for which they have to press the light bulb) and feedback. In order to improve the pedagogical success of the activity, the correct solution is presented automatically to the students if they fail to compose the correct price after 3 trials.

¹<http://www.gostai.com/products/jazz/urbi/>

²<http://processing.org>

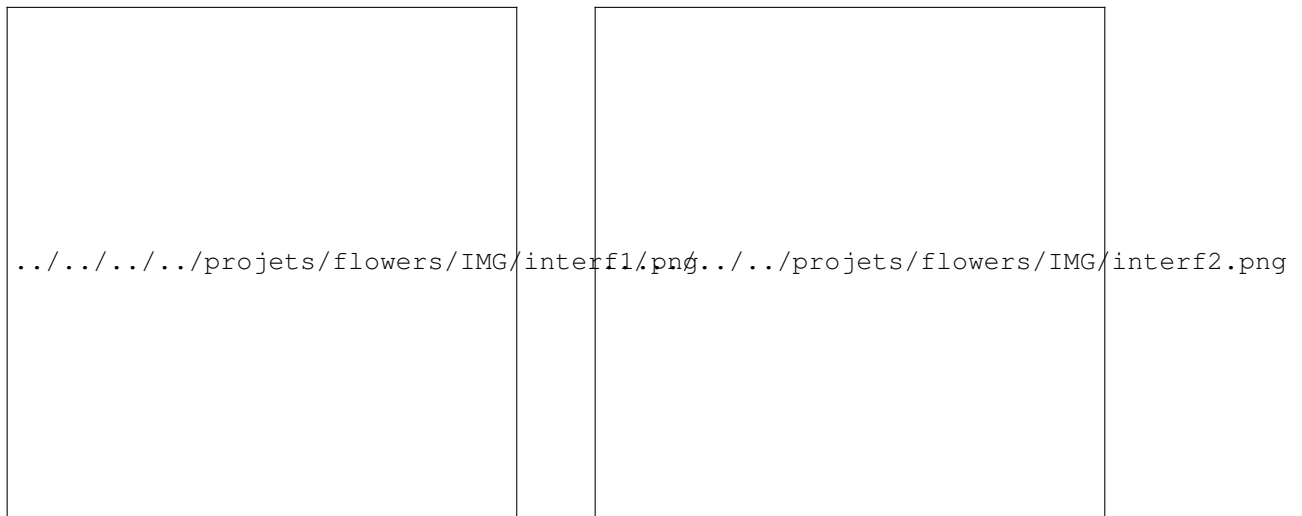


Figure 5. Example interface with four principal regions of interaction: wallet location, repository, object and price, and feedback location.

5.3.2. RLPark - Reinforcement Learning Algorithms in JAVA

Participant: Thomas Degris [correspondant].

RLPark is a reinforcement learning framework in Java. RLPark includes learning algorithms, state representations, reinforcement learning architectures, standard benchmark problems, communication interfaces for three robots, a framework for running experiments on clusters, and real-time visualization using Zephyr. More precisely, RLPark includes:

- Online Learning Algorithms: Sarsa, Expected Sarsa, Q-Learning, On-policy and off-policy Actor-Critic with normal distribution (continuous actions) and Boltzmann distribution (discrete action), average reward actor-critic, TD, TD(λ), GTD(λ), GQ(λ), TDC
- State Representations: tile coding (with no hashing, hashing and hashing with mumur2), Linear Threshold Unit, observation history, feature normalization, radial basis functions
- Interface with Robots: the Critterbot, iRobot Create, Nao, Puppy, Dynamixel motors
- Benchmark Problems: mountain car, swing-up pendulum, random walk, continuous grid world

An example of RLPark running an online learning experiment on a reinforcement learning benchmark problem is shown in Figure 6 .

RLPark was started in spring 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. RLPark is still actively used by RLAI. Collaborators and users include Adam White, Joseph Modayil and Patrick Pilarski (testing) from the University of Alberta.

RLPark has been used by Richard Sutton, a professor and iCORE chair in the department of computing science at the University of Alberta, for a demo in his invited talk *Learning About Sensorimotor Data* at the Neural Information Processing Systems (NIPS) 2011 ³. Patrick Pilarski used RLPark for live demos on television (Breakfast Television Edmonton, CityTV, June 5th, 2012) and at TEDx Edmonton on Intelligent Artificial Limbs⁴. So far, RLPark has been used in more than a dozens of publications (see <http://rlpark.github.com/publications.html> for a list).

³<http://webdocs.cs.ualberta.ca/~sutton/Talks/Talks.html#sensorimotor>

⁴<http://www.youtube.com/watch?v=YPc-Ae7zqSo>

RLPark has been ported to C++ by Saminda Abeyruwan, a student of the University of Miami (United States of America). The Horde architecture in RLPark has been optimized for GPU by Clément Gehring, a student of the McGill University in Montreal (Canada).

Future developments include the implementation of additional algorithms (the Dyna architecture, back propagation in neural networks, ...). A paper is under review for the JMLR Machine Learning Open Source Software. Documentation and tutorials are included on the RLPark web site ⁵. RLPark is licensed under the open source Eclipse Public License.



Figure 6. An example of an experiment in RLPark. Zephyr displays two views of a learned weight vector, an animation of the problem, the current policy distribution learned by the algorithm and the reward obtained by the algorithm. Videos are available at: <http://rlpark.github.com>.

5.3.3. DMP-BBO Matlab library

Participant: Freek Stulp [correspondant].

The `dmp_bbo` (Black-Box Optimization for Dynamic Movement Primitives) Matlab library is a direct consequence of the insight that black-box optimization outperforms reinforcement learning when using policies represented as Dynamic Movement Primitives. It implements several variants of the PI^{BB} algorithm for direct policy search. It is currently being used and extended by several FLOWERS members (Manuel Lopes, Clément Moulin-Frier) and external collaborators (Jonas Buchli, Hwangbo Jemin of ETH Zurich). This code was used for the following publications: [130], [127], [128].

⁵<http://rlpark.github.com>

5.3.4. Self-calibration BCI - Matlab library

Participants: Jonathan Grizou [correspondant], Iñaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer.

The Matlab software implements the algorithms described in [45]. It allows a robot to be instructed a new task by a human using communicative signals initially totally unknown to the robot. It is currently extended and improved in the context of EEG-based brain-machine interfaces (BMIs) [44].

It results in a BCI based control of sequential tasks with feedback signals that do not require any calibration process. As a by-product, the method provides an unsupervised way to train a decoder with the same performance of state-of-the-art supervised classifiers, while keeping the system operational and solving, with a lower performance during the first steps, the unknown task. The algorithm has been tested with online experiments (fig. 7), showing that the users were able to guide from scratch an agent to a desired position.



Figure 7. Results from the online BCI experiment for identifying the task. Evolution of the probability of the taught task for each subject and run

To improve the efficiency of the algorithm, we introduced a new planning method that uses the uncertainty in the signal-target estimation. This planner is inspired by exploration methods with exploration bonuses that allow guiding to reduce the uncertainty in an efficient way. We showed that trying to follow the best hypothesis

does not explore the space significantly to reduce uncertainty and thus identify the correct task. Only through an approach that plans how to reduce the uncertainty multiple steps ahead are we sure that the agent will reach states that can only be explained by the correct hypothesis.



Figure 8. Comparison between different exploration methods. Planning wrt. uncertainty in noth task and signal space is the most efficient method

5.3.5. *PROPRE: simulation of developmental concept formation using PYTHON*

Participant: Alexander Gepperth [correspondant].

This simulation software implements the algorithms described in [86], [83]. It is available online under the URL www.gepperth.net/downloads.html. The simulation is implemented in PYTHON for easy use, yet the time-critical core functions are written in C.

5.3.6. *pyStreamPlayer: synchronized replay of multiple sensor recordings and supplementary data*

Participant: Alexander Gepperth [correspondant].

This Python software is intended to facilitate the application of machine learning algorithms by avoiding to work directly with an embodied agent but instead with data recorded in such an agent. Assuming that non-synchronous data from multiple sensors (e.g., camera, Kinect, laser etc.) have been recorded according to a flexible format defined by the pyStreamPlayer architecture, pyStreamPlayer can replay these data while retaining the exact temporal relations between different sensor measurements. As long as the current task does not involve the generation of actions, this software allows to process sensor data as if it was coming from an agent which is usually considerably easier. At the same time, pyStreamPlayer allows to replay arbitrary supplementary information such as, e.g., object information, as if it was coming from a sensor. In this way, supervision information can be stored and accessed together with sensory measurements using an unified interface. pyStreamPlayer has been used to facilitate real-world object recognition tasks, and several of the major databases in this field (CalTech Pedestrian database, HRI RoadTraffic traffic objects database, CVC person database, KITTI traffic objects database) have been converted to the pyStreamPlayer format and now serve as a source of training and test data for learning algorithms.

pyStreamPlayer has been integrated into a ROS node as well, allowing the replay and transmission across networks of distributed processes.

5.3.7. Multimodal: framework around the NMF algorithm for multimodal learning

Participant: Olivier Mangin [correspondant].

The python code provides a minimum set of tools and associated libraries to reproduce the experiments on [53], together with the choreography datasets. The code, publicly available at <https://github.com/omangin/multimodal>, under the new BSD license, is primarily intended for reproduction of the multimodal learning experiment mentioned above. It is also expected that the public availability of the code encourages further experimentation by other scientists with data coming from other domains, thus increasing both the impact of the aforementioned publication and the knowledge on the algorithm behaviors. The nonnegative matrix factorization algorithm used in the experiments will also soon be included as a third party project to <http://scikit-learn.org>. Finally the code is currently being used by other members of the team and is expected to play an important role in further collaborations.

5.3.8. Tools for curiosity-driven learning on a robotic arm

Participants: Pierre Rouanet [correspondant], Clément Moulin-Frier.

This library is intended to provide tools for experimenting how curiosity model can facilitate learning of complex tasks such as manipulating objects. First, it provides high-level access to a robotic arm made of dynamixel motors: forward and inverse kinematics, demonstrations recording. Then it wraps the IMLE [77] library which we used for incremental and online learning of the sensorimotor mappings of the robot. Finally, it implements curiosity-driven learning based on the maximization of the learning progress. This modelling is based on recent works by Moulin-Frier and Oudeyer [54], [56], [55] proposing a probabilistic algorithmic architecture unifying various principles of developmental robotics such as motor babbling, goal babbling and curiosity-driven exploration. This architecture has already been successfully applied to model infant speech acquisition in the previously cited papers.

5.4. Software Platforms

5.4.1. Meka robot platform enhancement and maintenance

Participants: Antoine Hoarau [ADT Engineer Since Nov. 2012], Freek Stulp [Supervisor], David Filliat [Supervisor].

Autonomous human-centered robots, for instance robots that assist people with disabilities, must be able to physically manipulate their environment. There is therefore a strong interest within the FLOWERS team to apply the developmental approach to robotics in particular to the acquisition of sophisticated skills for manipulation and perception. ENSTA-ParisTech has recently acquired a Meka (cf. 9) humanoid robot dedicated to human-robot interaction, and which is perfectly fitted to this research. The goal of this project is to install state-of-the-art software architecture and libraries for perception and control on the Meka robot, so that this robot can be jointly used by FLOWERS and ENSTA. In particular, we want to provide the robot with an initial set of manipulation skills.

The goal is to develop a set of demos, which demonstrate the capabilities of the Meka, and provide a basis on which researchers can start their experiments.

The platform is evolving as the software (Ubuntu, ROS, our code) is constantly updated and requires some maintenance so less is needed for later. A few demos were added, as the hand shaking demo, in which the robot detects people via kinect and initiates a hand shake with facial expressions. This demo has been used to setup a bigger human robot interaction experiment, currently tested on subjects at Ensta (cf. 10). The stacking cups demo is more of a manipulation and vision demo : the robot detects its cups by their shape and color, and tries to make a tower with it (cf. 12). This demo has required to manually update the old pr2 tabletop object detector to the new ROS version, create a tool to semi-automatically calibrate the extrinsic parameters of the kinect, and different set of tools to catch objects from different angles (waiting for moveit to be fully integrated). Finally, we've seen that the robot itself also needs some maintenance; some components broke (a finger tendon), a welding got cold (in the arm) and a few cables experienced fatigue (led matrix and cameras) (cf. 11).

I've also given a talk at Humanoid 2013 at Atlanta and the University of Texas at Austin for my participation on [61].

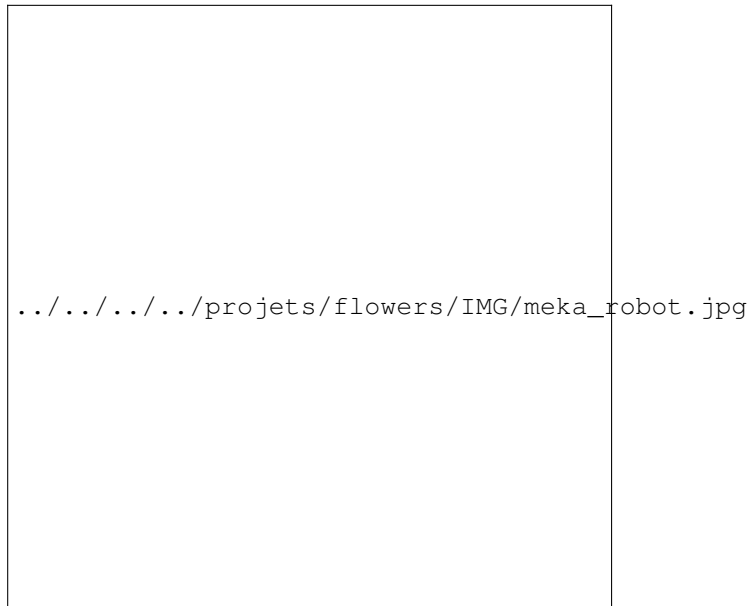


Figure 9. The Meka robot platform acquired by ENSTA ParisTech

5.4.2. Experiment platform for multiparameters simulations

Participants: Fabien Benureau, Paul Fudal.



Figure 10. Hand shake demo visualized on Rviz (ROS)



Figure 11. Maintenance is required on the robot



Figure 12. The stacking cup demo (video available at <http://cogrob.ensta-paristech.fr>)

Simulations in robotics have many shortcomings. At the same time, they offer high customizability, rapidity of deployment, absence of failure, consistency across time and scalability. In the context of the PhD work of Fabien Benureau, it was decided to investigate hypothesis first in simulation before moving to real hardware. In order to be able to test a high number of different hypothesis, we developed a software platform that would scale to the computing resource available.

We designed simple continuous simulations around a of-the-shelf 2D physic engine and wrote a highly modular platform that would automatically deploy experiments on cluster environments, with proper handling of dependencies; our work investigate transfer learning, and some experiments's input data is dependent of the results of another.

So far, this platform and the university cluster has allowed to conduct thousands of simulations in parallel, totaling more than 10 years of simulation time. It has led us to present many diverse experiments in our published work [40], each repeated numerous times. It has allowed us to conduct a multi-parameter analysis on the setup, which led to new insights, which are being presented in a journal article to be submitted in the beginning of this year.

Because of its high modularity, this platform is proving to be highly flexible. We are currently adaptating it to a modified, cluster-ready, version of the V-REP simulator. Those simulations will serve to back ones on similar real-world hardware that are currently setup.

We have released the platform and the complete experiments code when we published the results of [40], allowing to reproduce the results of the paper, and will continue to do so with each published work.

5.4.3. PyPot

Participants: Pierre Rouanet [correspondant], Matthieu Lapeyre.

PyPot is a framework developed to make it easy and fast to control custom robots based on dynamixel motors. This framework provides different levels of abstraction corresponding to different types of use. More precisely, you can use PyPot to:

1. directly control robot's motors through a USB2serial device,
2. define the structure of your particular robot and control it through high-level commands,
3. define primitives and easily combine them to create complex behavior.

PyPot has been entirely written in Python to allow for fast development, easy deployment and quick scripting by non-necessary expert developers. It can also benefit from the scientific and machine learning libraries existing in Python. The serial communication is handled through the standard library and thus allows for rather high performance (10ms sensorimotor loop). It is cross-platform and has been tested on Linux, Windows and Mac OS.

PyPot is part of the Poppy project (<http://www.poppy-project.org>) and has been released under an open source license GPL V3. More details are available on PyPot website: <https://github.com/poppy-project/pypot>

5.5. Experimental Setups

5.5.1. Experimental Platform for User Study of Curiosity-driven Exploration

Participants: Pierre Rouanet [correspondant], Jonathan Grizou, Brice Miard, Julie Golliot.

This platform has been developed to investigate curiosity-driven behaviors and more precisely how humans explore new sensori-motor spaces. It consists in several simple games where users control a 2D/3D shape with the movements of their body. They have to discover the mapping between their movements and a shape displayed on the screen and learn how to make the controlled shape match the target one (fig 13).



Figure 13. A screenshot representing the game interface as seen by the user.

The software is entirely written in Python. It includes a Kinect wrapper allowing the access of 3D position of tracked skeleton joints. It provides a framework for creating new games based on the 2D drawing library (pygame). It also includes a web server used to display game instructions, cut-scene videos and questionnaire.

5.6. Visualization Tools

5.6.1. Zephyr - Realtime Visualization in JAVA

Participant: Thomas Degris [correspondant].

Zephyr is a software to visualize numeric variables and data structure in real time and at different time scale. Zephyr is practical because it requires only minimal changes in the code: it uses Java reflexivity to automatically detect variables in the code to monitor and data structure with an associated dedicated view. Zephyr can easily be extended with new plugins because it is based on the popular Eclipse Rich Client Platform. Consequently, Zephyr takes advantage of an already existing and fully operational Eclipse plugins for many of its functionalities. Finally, Zephyr is distributed with a Java python virtual machine named Jython and a lisp implementation named Clojure. An example of a Zephyr screen is shown in Figure 14 .

Zephyr was started in fall 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. Zephyr is still actively used by RLAI. Users include Adam White, Joseph Modayil and Patrick Pilarski from the University of Alberta. Zephyr has been registered on the Eclipse marketplace since October 2011. Documentation about Zephyr is included on its website: <http://zephyrplugins.github.com>. Zephyr is licensed under the open source Eclipse Public License.

5.6.2. Experimental Setups for User Study of Alignment in Asymmetric Interactions

Participants: Jonathan Grizou [correspondant], Chloé Rozenbaum, Manuel Lopes, Katharina Rohlfing, Pierre-Yves Oudeyer.



Figure 14. Left: Zephyr showing the different steps of a video processing pipeline in real-time. Right: Zephyr showing different data structure and variables of a reinforcement learning agent at different time scale. A video is available at: <http://zephyrplugins.github.com>.

This platform has been developed to investigate alignment in asymmetric interactions. We consider a remote construction task, where one user (user A) knows what to build but do not have access to the construction site while its partner (user B) is at the site but do not know what to do. By constraining the communicative channel between the two partners, we study how, and if, they will agree on a similar set of signals to convey information and what type of information they tend to produce.

The experimental setup consist of box with button, a video recording system and two screens. User A can send signals to user B by pressing buttons (fig. 15). Signals are displayed on a screen (fig. 15) at user B side. User A is not aware of what is displayed on user B screen, neither user B is aware of the relation between button presses and screen events. The video of user B construction scene is streamed to a screen at user B side.

The task consist of bulding arbitrary construction (fig. 15) using colored toy bricks (fig. 15).

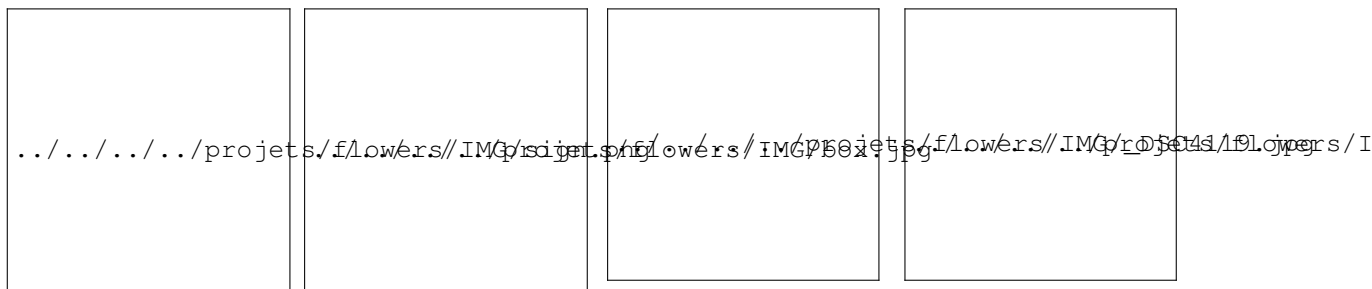


Figure 15. Three examples of sign displayed on the learner screen; The box and the button use as an interface for the teacher to communicate with the learner; Examples of construction presented to the teacher.

5.7. Hardware

5.7.1. Poppy Platform

Participants: Matthieu Lapeyre [correspondant], Pierre Rouanet, Jonathan Grizou, Pierre-Yves Oudeyer [supervisor].

5.7.1.1. Main goals :

No current platform (Nao [87], Darwin Op [88], Nimbro Op [126], HRP-2, ...) does offer both a adapted morphology in the sense of allowing physical interaction (safe, compliant, playful) and optimized for walking. So to explore these challenges we have decided to build a new bio-inspired humanoid robotic platform, called Poppy, which provides some of the software and hardware features needed to explore both social interaction and biped locomotion for personal robot. It presents the following main features to make it an interesting platform to study how the combination of morphology and social interaction can help the learning:

- Design inspired from the study of the anatomy of the human body and its bio-mechanic
- Dynamic and reactive: we try to keep the weight of the robot as low as possible (geometry of the pieces and smaller motors)
- Social interaction: screen for communication and permits physical interaction thanks to compliance
- Study of the morphology of the leg to improve the biped walking
- Practical platform: low cost, ease of use and easy to reproduce

5.7.1.2. Overview :

Poppy platform (Figure 16) is a humanoid, it is 84cm tall for 3 kg. It has a large sensor motors space including 25 dynamical motors (MX-28 and AX-12), force sensors under its feet and some extra sensors in the head: 2 HD-wide angle-cameras, stereo-micros and an inertial central unit (IMU 9DoF) plus a large LCD Screen (4 inch) for visual communication (e.g. emotions, instructions or debug). The mechanical parts were designed and optimized to be as light as possible while maintaining the necessary strength. For this, the choice of a lattice beam structure manufactured with 3Dprinting polyamide was used.

The poppy morphology is designed based on the actual human body. We have deeply studied the biomechanics of the human body and have extracted some interesting features for humanoid robotics. This inspiration is expressed in the whole structure (e.g. the limb proportions) and in particular in the trunk and legs.



Figure 16. a. Global view of the Poppy platform. b. Zoom on legs design

Poppy uses the bio-inspired trunk system introduced by Acroban [101]. These five motors allow it to reproduce the main changes brought by the human spine. This feature allows the integration of more natural and fluid motion while improving the user experience during physical interactions. In addition, the spine plays a fundamental role in bipedal walking and postural balance by actively participating in the balancing of the robot.

The legs were designed to increase the stability and agility of the robot during the biped walking by combining bio-inspired, semi-passive, lightweight and mechanical-computation features. We will now describe two examples of this approach:

The architecture of the hips and thighs of Poppy uses biomechanical principles existing in humans. The human femur is actually slightly bent at an angle of about 6 degrees. In addition, the implantation of the femoral head in the hip is on the side. This results in a reduction of the lateral hip movement needed to move the center of gravity from one foot to another and a decrease in the lateral falling speed. In the case of Poppy, the inclination of its thighs by an angle of 6 degrees causes a gain of performance of more than 30% for the two above mentioned points.

Another example is Poppy's feet. Poppy has the particularity of having small feet compared to standard humanoids. It has humanly proportioned feet (ie about 15% of its total size). It is also equipped with compliant toes joints (see Figure 17 .a). We believe that this feet involve two keys features to obtain a human-like and efficient walking gait. However, that raises problems regarding balance because the support polygon is reduced. We decided to add pressure sensors under each foot in order to get accurate feedback of the current state of the robot (see Figure 17 .b).



Figure 17. Poppy feet use actual children shoes combine with a compliant feet, toes (a.) and pressure sensors (b.)

5.7.1.3. Open source release :

To allow the distribution in the robotic community, we have decided to make Poppy an open platform. So the software and the hardware are open source. They have each a repository. The PyPot library is under GPLv3 license and is available on a bitbucket repository (<https://github.com/poppy-project/pypot>). The hardware, Solidworks files and STL needed to print the robot are available under a Creative Commons BY+SA+NC license on a private GrabCAD Workbench repository. People can request access to the hardware repository on the Poppy website (<http://www.poppy-project.org/open-platform/>).

The platform is currently under beta-testing meaning that we let the community grows little by little to ensure a good support of interesting projects.

For now, there are about 200 people on the GrabCAD project. They have access to all files needed to print the robot. There is also about 60 beta testers. They have access to a private section on the website with documentation and a forum for support.

Several of them are already doing a great work both by reporting bugs and managing to build the robot outside the lab. We are trying to work closely with them as they are a great source of feedback to improve the platform before a more wide distribution.

5.7.1.4. Impact in the community

Poppy has been release open source the 15/10/2013. The announcement has been done by Pierre Yves Oudeyer during the Lift 2013 conference. To prepare this event, the website and a overview video were made. The video, accessible here (<http://vimeo.com/76917854>) has reached about 40K views until now.

A part of the audience are technology-enthusiast people (about 50% on grabCAD), interested by the fact Poppy is 3D printable and so, highly customizable. More interesting, we received a large number of beta request from various applications domains (see 18) around the world (see 19).



Figure 18. The 60 selected beta testers profiles chart.

5.7.1.4.1. Education:

We receive a lot of request from Fablab around the world (France, USA, New-Zeland, South Africa,...). All have great educational projects for teaching robotics, mechanics and computer science to children. Some of them are close beta testers and we have weekly interactions. In the same topic, several schools, engineering schools and universities showed interest to use Poppy as experimental support. One of our close beta tester is a Bordeaux high school. They are planning to use Poppy as support for mechanics, computer sciences but also architecture or philosophy.

5.7.1.4.2. Art:

A current art project is under construction. A residence with an artist, a dancer and us will take place on the 24/02/2014 to the 05/03/2014. It comes to artistically invest humanoid robotics and thereby examine the relationship of the body to the digital world. The encounter between art and science generates potential new ideas for both disciplines who find themselves at the crossroads of questions relating to the gesture, movement and body. The Poppy Project is focused on morphological and motor aspects in a context human-robot interface. The look of an artist and the movements of a dancer are testing this interface with an unprecedented and direct manere.



Figure 19. Where are located the beta testers?

5.7.1.4.3. Research:

A large number of researchers showed interest in the platform. Most of them are interested to use it as a experimental tool. They want to address challenges such as balance and walking control, use of force controlled motors, explore human-robot interactions. On this last topic, the Bristol robotic lab target to use tele-operation to investigate what factors are important in terms of appearance and behaviour for credible and trustworthy interaction. For this purpose, they will develop functional hands for gesture and grasping task.

5.7.1.5. Next step:

We are currently working with beta testers on several improvements to make Poppy more accessible, more easy to use and more polyvalent. Two internship students will arrive on the project to work on the embedded electronic and on the feet design. We are targeting to release the final version during the summer 2014. Also we are thinking about the creation of a "Poppy pack" including all necessary components and tools to easily build the robot.

Another very important point is the community management. We are currently adding new collaborative tools on the Poppy website. These tools are tested with beta testers. The challenge is to offer the good tools to provide an efficient support to future users and to encourage people to contribute. This work is done in collaboration with Stephane Ribas (D2T inria Grenoble).

GEOSTAT Project-Team

5. Software and Platforms

5.1. Fluex

Participants: Denis Arrivault [correspondant], Rémi Paties, Hussein Yahia, Joel Sudre.

- Denis Arrivault has joined the team for a complete refoundation, rewriting, generalization and diffusion of the FluidExponents software, now called Fluex. FluidExponents is a software implementation of the MMF, presently written in Java, in a cooperative development mode on the Inria GForge, deposited at APP in 2010. Denis Arrivault has delivered the first Fluex package in December 2013, consisting of a core implementation under Gforge of the Microcanonical Multiscale Formalism in the form of C++ classes, for 1D, 2D 3D and 3D+t general signals. Fluex is in the process of being deposited in 2014. The Fluex project is carried on in 2014 by Rémi Paties. Contact: denis.arrivault@inria.fr, remi.paties@inria.fr
- A matlab code for the speech GCI detection algorithm has been made publicly available on the GeoStat website.

HIEPACS Project-Team

5. Software and Platforms

5.1. Introduction

We describe in this section the software that we are developing. The first list will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

5.2. MaPHyS

Participant: Emmanuel Agullo [corresponding member].

MaPHyS (Massively Parallel Hybrid Solver) is a software package whose prototype was initially developed in the framework of a PhD thesis and further consolidated first thanks to the ANR-CIS Solstice funding and the Inria Parscali ADT. This parallel linear solver couples direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

The **MaPHyS** package is very much a first outcome of the research activity described in Section 3.3 . Finally, **MaPHyS** is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We foresee to either embed in **MaPHyS** some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3 .

MaPHyS can be found at <http://maphys.gforge.inria.fr>.

5.3. PaStiX

Participant: Pierre Ramet [corresponding member].

Complete and incomplete supernodal sparse parallel factorizations.

PaStiX (Parallel Sparse matriX package) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The **PaStiX** library uses the graph partitioning and sparse matrix block ordering package Scotch. **PaStiX** is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists in defining an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughs in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.

PaStiX is publicly available at <http://pastix.gforge.inria.fr> under the Inria CeCILL licence.

5.4. HIPS

Participant: Pierre Ramet [corresponding member].

Multilevel method, domain decomposition, Schur complement, parallel iterative solver.

HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

The key point of the methods implemented in **HIPS** is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. **HIPS** exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file ; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, **HIPS** is a software library that provides several methods to build an efficient preconditioner in almost all situations.

HIPS is publicly available at <http://hips.gforge.inria.fr> under the Inria CeCILL licence.

5.5. LBC2

Participant: Aurélien Esnard [corresponding member].

LBC2 (Load-Balancing for Code Coupling) is a library providing different methods for partitioning or repartitioning graphs and hypergraphs. These methods are designed for effective communications during dynamic load balancing with a variable number of processors or coupled code interactions. Repartitioning is achieved with an enriched graph model partitioned using third-party partitioning tools, as Scotch, PaToH, METIS, Zoltan or Mondrian.

The framework is publicly available at Inria Gforge as a part of the MPICPL framework: <http://mpicpl.gforge.inria.fr>.

5.6. MPICPL

Participant: Aurélien Esnard [corresponding member].

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the *mpicplrun* tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the *mpicplrun* tool.

MPICPL was developed by HIEPACS for the purpose of the ANR NOSSI. It uses advanced features of MPI2 standard. The framework is publicly available at Inria Gforge: <http://mpicpl.gforge.inria.fr>.

5.7. ScalFMM

Participant: Olivier Coulaud [corresponding member].

ScalFMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There is two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, **ScalFMM** is using CMAKE as a compiler/installer tool. Even if **ScalFMM** is written in C++ it will support a C and fortran API soon.

The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve at Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

The **ScalFMM** package is available at <http://scalform.gforge.inria.fr>

5.8. ViTE

Participant: Mathieu Faverge [corresponding member].

Visualization, Execution trace

ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications. It is an open source software licenced under CeCILL-A.

The **ViTE** software is available at <http://vite.gforge.inria.fr> and has been developed in collaboration with the Inria Bordeaux - Sud-Ouest SED team, Telecom SudParis and Inria Grenoble.

In the same context we also contribute to the EZtrace and GTG libraries in collaboration with F. Trahay from Telecom SudParis. EZTrace (<http://eztrace.gforge.inria.fr>) is a tool that aims at generating automatically execution trace from HPC programs. It generates execution trace files thanks to the GTG library (<http://gtg.gforge.inria.fr>) that can be later interpreted by visualization tools such as **ViTE**.

5.9. Other software

For the materials physics applications, a lot of development will be done in the context of ANR projects (**NOSSI** and **OPTIDIS**, see Section 4.1) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- **FAST**

Participant: Olivier Coulaud [corresponding member].

FAST is a linear response time dependent density functional program for computing the electronic absorption spectrum of molecular systems. It uses an $O(N^3)$ linear response method based on finite numerical atomic orbitals and deflation of linear dependence in atomic orbital product space. This version is designed to work with data produced by the SIESTA DFT code. The code produces as principal output a numerical absorption spectrum (complex part of the polarisability, loosely called the polarisability below) and a list of transition energies and oscillator strengths deduced from fitting Lorentzians to the numerical spectrum. Considering the absence of hybrid functionals in SIESTA and that concerning calculation of spectra, generalized gradient Hamiltonians are not usually considered to be notably better than the local density approximation, the present release of **FAST** works only with LDA, which despite its limitations, has provided useful results on the systems to which the present authors have applied it. The **FAST** library is available at <http://people.bordeaux.inria.fr/coulaud/Softwares/FAST/index.html>.

- **OptiDis**

Participant: Olivier Coulaud [corresponding member].

OptiDis is a new code for large scale dislocation dynamics simulations. its aim is to simulate real life dislocation densities (up until $5 \cdot 10^{22}$ dislocations/ m^{-2}) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation on irradiated zirconium. Zirconium alloys is the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking of channeling mechanism, which mean to stick with the reality, more than tens of thousands of induced loops so 10^8 degrees of freedom in the simulation.

The code is based on Numodis code developed at CEA Saclay and the **ScaLFMM** library developed in our Inria project. The code is written in C++ language and using the last features of C++11. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rise. In order to achieve that, we use different level of parallelism. First of all, the simulation box is spread over MPI process, we then use a thinner level for threads, dividing the domain using an Octree representation. All these parts are driven by the **ScaLFMM** library. On the last level our data are stored in an adaptive structure absorbing dynamic of this kind of simulation and handling well task parallelism.

The two following packages are mainly designed and developed in the context of a US initiative led by ICL and to which we closely collaborate through the associate team **MORSE**.

- **PLASMA**

Participant: Mathieu Favergé [corresponding member].

The **PLASMA** (Parallel Linear Algebra for Scalable Multi-core Architectures) project aims to address the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multi-core architectures.

The **PLASMA** ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures.

The development of programming models that enforce asynchronous, out of order scheduling of operations is the concept used as the basis for the definition of a scalable yet highly efficient software framework for Computational Linear Algebra applications.

The **PLASMA** library is available at <http://icl.cs.utk.edu/plasma>.

- **PaRSEC/DPLASMA**

Participant: Mathieu Faverge [corresponding member].

PaRSEC Parallel Runtime Scheduling and Execution Controller, is a generic framework for architecture aware scheduling and management of micro-tasks on distributed many-core heterogeneous architectures. Applications we consider can be expressed as a Direct Acyclic Graph of tasks with labeled edges designating data dependencies. DAGs are represented in a compact problem-size independent format that can be queried on-demand to discover data dependencies in a totally distributed fashion. **PaRSEC** assigns computation threads to the cores, overlaps communications and computations and uses a dynamic, fully-distributed scheduler based on architectural features such as NUMA nodes and algorithmic features such as data reuse.

The framework includes libraries, a runtime system, and development tools to help application developers tackle the difficult task of porting their applications to highly heterogeneous and diverse environment.

DPLASMA (Distributed Parallel Linear Algebra Software for Multicore Architectures) is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. **DPLASMA** achieves this objective through the state of the art **PaRSEC** runtime, porting the **PLASMA** algorithms to the distributed memory realm.

The **PaRSEC** runtime and the **DPLASMA** library are available at <http://icl.cs.utk.edu/parsec>.

LFANT Project-Team

5. Software and Platforms

5.1. Pari/Gp

Participants: Karim Belabas [correspondent], Bill Allombert, Henri Cohen, Andreas Enge.

<http://pari.math.u-bordeaux.fr/>

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- PARI is a C library, allowing fast computations.
- GP is an easy-to-use interactive shell giving access to the PARI functions.
- gp2c, the GP-to-C compiler, combines the best of both worlds by compiling GP scripts to the C language and transparently loading the resulting functions into GP; scripts compiled by gp2c will typically run three to four times faster.
- Version of PARI/GP: 2.5.5
- Version of gp2c: 0.0.8
- License: GPL v2+
- Programming language: C

5.2. GNU MPC

Participants: Andreas Enge [correspondent], Mickaël Gastineau [CNRS], Philippe Théveny [INRIA project-team ARIC], Paul Zimmermann [INRIA project-team CARAMEL].

<http://mpc.multiprecision.org/>

GNU MPC is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as GNUMPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran front ends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

2012 has seen the first release of the major version 1.0.

- Version: 1.0.1 *Fagus silvatica*
- License: LGPL v3+
- ACM: G.1.0 (Multiple precision arithmetic)
- AMS: 30.04 Explicit machine computation and programs
- APP: Dépôt APP le 2003-02-05 sous le numéro IDDN FR 001 060029 000 R P 2003 000 10000
- Programming language: C

5.3. MPFR CX

Participant: Andreas Enge.

<http://mpfrcx.multiprecision.org/>

MPFRGX is a library for the arithmetic of univariate polynomials over arbitrary precision real (MPFR) or complex (MPC) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Version: 0.4.2 *Cassava*
- License: LGPL v2.1+
- Programming language: C

5.4. CM

Participant: Andreas Enge.

<http://cm.multiprecision.org/>

The CM software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications. For the implemented algorithms, see [8].

- Version: 0.2 *Blindhühnchen*
- License: GPL v2+
- Programming language: C

5.5. AVIsogenies

Participants: Damien Robert [correspondent], Gaëtan Bisson, Romain Cosset [INRIA project-team CARAMEL].

<http://avisogenies.gforge.inria.fr/>

AVISOGENIES (Abelian Varieties and Isogenies) is a MAGMA package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of (ℓ, ℓ) -isogenies between Jacobian varieties of genus-two hyperelliptic curves over finite fields of characteristic coprime to ℓ ; practical runs have used values of ℓ in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Version: 0.6
- License: LGPL v2.1+
- Programming language: Magma

5.6. APIP

Participant: Jérôme Milan.

<http://www.lix.polytechnique.fr/~milanj/apip/apip.xhtml>

APIP, Another Pairing Implementation in PARI, is a library for computing standard and optimised variants of most cryptographic pairings.

The following pairings are available: Weil, Tate, ate and twisted ate, optimised versions (à la Vercauteren–Hess) of ate and twisted ate for selected curve families.

The following methods to compute the Miller part are implemented: standard Miller double-and-add method, standard Miller using a non-adjacent form, Boxall et al. version, Boxall et al. version using a non-adjacent form.

The final exponentiation part can be computed using one of the following variants: naive exponentiation, interleaved method, Avanzi-Mihalescu's method, Kato et al.'s method, Scott et al.'s method.

- Version: 2012-10-17
- License: GPL v2+
- Programming language: C with libpari

5.7. CMH

Participants: Andreas Enge, Emmanuel Thomé [INRIA project-team CAMEL].

<http://cmh.gforge.inria.fr/>

CMH computes Igusa class polynomials, parameterising two-dimensional abelian varieties (or, equivalently, Jacobians of hyperelliptic curves of genus 2) with given complex multiplication.

- Version: development snapshot
- License: GPL v3+
- Programming language: C

5.8. Cubic

Participant: Karim Belabas.

<http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz>

CUBIC is a stand-alone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the PARI library. The algorithm has quasi-linear time complexity in the size of the output.

- Version: 1.2
- License: GPL v2+
- Programming language: C

5.9. Euclid

Participant: Pierre Lezowski.

<http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php>.

Euclid is a program to compute the Euclidean minimum of a number field. It is the practical implementation of the algorithm described in [41]. Some corresponding tables built with the algorithm are also available. Euclid is a stand-alone program depending on the PARI library.

- Version: 1.0
- License: LGPL v2+
- Programming language: C

5.10. KleinianGroups

Participant: Aurel Page.

<http://www.normalesup.org/~page/Recherche/Logiciels/logiciels.html>

KLEINIANGROUPS is a Magma package that computes fundamental domains of arithmetic Kleinian groups.

- Version: 1.0
- License: GPL v3+
- Programming language: Magma

MAGIQUE-3D Project-Team

5. Software and Platforms

5.1. Hou10ni

Participant: Julien Diaz [correspondant].

5.1.1. Hou10ni-Time-Domain

This software, written in FORTRAN 90, simulates the propagation of acoustic waves in heterogeneous 2D and 3D media. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM). The 2D version of the code has been implemented in the Reverse Time Migration (RTM) software of TOTAL in the framework of the Ph.D thesis of Caroline Baldassari. The 2D code allows for the use of meshes composed of cells of various order (p -adaptivity in space). For the time discretization, we used the local time stepping strategy described at section 3.2, item **High-Order Schemes in Space and Time** which permits not only the use of different time-step, but also to adapt the order of the time-discretization to the order of each cells (hp -adaptivity in time).

The main competitors of Hou10ni are codes based on Finite Differences, Spectral Element Method or other Discontinuous Galerkin Methods (such as the ADER schemes). During her Ph.D thesis, Caroline Baldassari compared the solution obtained by Hou10ni to the solution obtained by a Finite Difference Method and by a Spectral Element Method (SPECFEM). To evaluate the accuracy of the solutions, we have compared them to analytical solutions provided by the codes Gar6more (see below). The results of these comparisons are: a) that Hou10ni outperforms the Finite Difference Methods both in terms of accuracy and of computational burden and b) that its performances are similar to Spectral Element Methods. Since Hou10ni allows for the use of meshes based on tetrahedrons, which are more appropriate to mesh complex topographies, and for the p -adaptivity, we decided to implement it in the RTM code of TOTAL. Of course, we also used these comparisons to validate the code.

5.1.2. Hou10ni-Frequency-Domain

Recently, we have extended the 2D version of Hou10ni for computing the solution of the harmonic wave equation (Helmholtz), in the framework of the PhD thesis of Élodie Estécahandy. This new version is able to deal with both acoustic and elastodynamic media, but also to model elastoacoustic problems. The surfaces between the different media can be approximated by curved elements. We can use up to P^{15} elements when dealing with curved elements and element of arbitrary order (with of course a limitation depending on the machine precision) when dealing with non-curved elements. The construction of the global matrix is perform using OpenMP and the extension to hybrid MPI/OpenMP parallelism is on development. This code has been also implemented in a solver which determine the shape of an elastic obstacle from the knowledge of its scattered field.

The 3D version of Hou10ni-Frequency-Domain is under development. The code is now able to solve acoustic problems up to P^3 elements. It has been parallelized using MPI and is able to deal with partitioned meshes. Preliminary tests have been performed up to 16.000.000 unknowns. We are now considering the following features : hybrid MPI/OpenMP parallelism; extension to arbitrary polynomial degrees; extension to elastodynamic.

5.2. Gar6more2D and Gar6more3D

Participant: Julien Diaz [correspondant].

This codes compute the analytical solution of problems of waves propagation in two layered 3D media such as-acoustic/acoustic- acoustic/elastodynamic- acoustic/porous- porous/porous, based on the Cagniard-de Hoop method.

See also the web page <http://web.univ-pau.fr/~jdiaz1/software.html>.

The main objective of these codes is to provide reference solutions in order to validate numerical codes. They have been already used by J. Tromp and C. Morency to validate their code of poroelastic wave propagation [96]. They are freely distributed under a CECILL license and can be downloaded on the website <http://web.univ-pau.fr/~jdiaz1/software.html>. As far as we know, the main competitor of this code is EX2DELDEL (available on <http://www.spice-rtn.org>), but this code only deals with 2D acoustic or elastic media. Our codes seem to be the only ones able to deal with bilayered poroelastic media and to handle the three dimensional cases.

- ACM: J.2
- AMS: 34B27 35L05 35L15 74F10 74J05
- Programming language: Fortran 90

5.3. Montjoie

Participant: Marc Duruflé [correspondant].

Montjoie is a code developed by Marc Duruflé with contributions of students, including Juliette Chabassier during her PhD. It provides a C++ framework for solving partial differential equations on unstructured meshes with finite element-like methods (continuous finite element, discontinuous Galerkin formulation, edge elements and facet elements). The handling of mixed elements (tetrahedra, prisms, pyramids and hexahedra) has been implemented for these different types of finite elements methods in the context of Morgane Bergot's PhD. Several applications are currently available : wave equation, elastodynamics, aeroacoustics, Maxwell's equations. In 2013, an implementation of non-linear 1-D Maxwell's equations (non-linear Kerr effect) has been added as well as various 1-D non-linear Schrödinger-like equations. Thin layer models are also available for Maxwell's equations and elastodynamics.

See also the web page <http://montjoie.gforge.inria.fr>.

MAGNOME Project-Team

5. Software and Platforms

5.1. Magus: Genome exploration and analysis

Participants: David James Sherman [correspondant], Pascal Durrens, Natalia Golenetskaya, Florian Lajus, Xavier Calcas.

The MAGUS genome annotation system integrates genome sequences and sequences features, *in silico* analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for simultaneous annotation of related genomes through the use of protein families identified by *in silico* analyses; this results in an n -fold increase in curation speed, compared to curation of individual genes. This allows us to maintain standards of high-quality manual annotation while efficiently using the time of volunteer curators. For more information see the MAGUS Gforge web site. ¹ MAGUS 1.x is mature software used since 2006 by our collaboration partners. MAGUS 2.0 is developed in an Inria Technology Development Action (ADT) with an open-source license and is being deposited with the APP.

5.2. Pantograph: Inference of metabolic networks

Participants: David James Sherman [correspondant], Pascal Durrens, Nicolás Loira, Anna Zhukova.

Pantograph is a software tool for inferring whole-genome metabolic models for eukaryote cell factories. It is based on metabolic scaffolds, abstract descriptions of reactions and pathways on which inferred reactions are hung are eventually connected by an interactive mapping and specialization process. Scaffold fragments can be repeatedly used to build specialized subnetworks of the complete model. A novel feature of Pantograph is that it uses expert knowledge implicitly encoded in the scaffold's gene associations, and explicitly transfers this knowledge to the new model. Pantograph is available under an open-source license. For more information see the Pantograph Gforge web site. ²

5.3. MetaModGen: Generalizing Metabolic Models

Participants: Anna Zhukova [correspondant], David James Sherman.

The metabolic model generalization and navigation software allows a human expert to explore a metabolic model in a layered manner. The software creates an on-line semantically zoomable representation of a model submitted by the user in SBML ³ format. The most general view represents the compartments of the model; the next view shows the visualization of generalized versions of reactions and metabolites in each compartment (see section 6.3); and the most detailed view visualizes the initial model with the generalization-based layout (where similar metabolites and reactions are placed next to each other). Zoomable representation is implemented using the Leaflet⁴ JavaScript library for mobile-friendly interactive maps. Users can click on reactions and compounds to see the information about their annotations. An example of a zoomable representation of the peroxisome compartment of *Y. lipolytica* is available at <http://metamogen.gforge.inria.fr/map.html>.

5.4. BioRica: Multi-scale Stochastic Modeling

Participants: David James Sherman [correspondant], Rodrigo Assar Cuevas, Joaquin Fernandez.

¹<http://magus.gforge.inria.fr>

²<http://pathtastic.gforge.inria.fr>

³<http://sbml.org>

⁴<http://leafletjs.com>

BioRica is a high-level modeling framework integrating discrete and continuous multi-scale dynamics within the same semantics field. A model in BioRica node is hierarchically composed of nodes, which may be existing models. Individual nodes can be of two types:

- Discrete nodes are composed of states and transitions described by guarded events. Behavior can be stochastic (defined by the likelihood that an event fires when activated) and timed (defined by the delay between an event's activation and the moment that its transition occurs).
- Continuous nodes are described by ODE systems, potentially a hybrid system whose internal state flows continuously while having discrete jumps.

The system has been implemented as a distributable software package. The BioRica compiler reads a specification for hierarchical model and compiles it into an executable simulator. The modeling language is a stochastic extension to the AltaRica⁵ Dataflow language, inspired by work of Antoine Rauzy. Input parsers for SBML 2 version 4 are currently being validated. The compiled code uses the Python runtime environment and can be run stand-alone on most systems. For more information see the BioRica Gforge web site.⁶ BioRica was developed as an Inria Technology Development Action (ADT) with an open-source license and is deposited with the APP.

5.5. Génolevures On Line: Comparative Genomics of Yeasts

Participants: Pascal Durrens [correspondant], Natalia Golenetskaya, Tiphaine Martin, David James Sherman.

The Génolevures online database provides tools and data for exploring the annotated genome sequences of more than 20 genomes, determined and manually annotated by the Génolevures Consortium to facilitate comparative genomic studies of hemiascomycetous yeasts. Data are presented with a focus on relations between genes and genomes: conservation of genes and gene families, speciation, chromosomal reorganization and synteny. The Génolevures site includes a private collaboration area for specific studies by members of its international community. The contents of the knowledge base are expanded and maintained by the CNRS through GDR 2354 Génolevures, and full data may be downloaded from the site. Génolevures online uses our open-source MAGUS system for genome navigation, with project-specific extensions developed by David Sherman, Pascal Durrens, and Tiphaine Martin; these extensions are not made available due to uncertainty about intellectual property rights. For more information see the Génolevures web site.⁷

5.6. Inria Bioscience Resources

Participants: Olivier Collin [correspondant], Frédéric Cazals, Mireille Régnier, Marie-France Sagot, Hélène Touzet, Hidde De jong, David James Sherman, Marie-Dominique Devignes, Dominique Lavenier.

Inria Bioscience Resources is a portal designed to improve the visibility of bioinformatics tools and resources developed by Inria teams. This portal will help the community of biologists and bioinformaticians understand the variety of bioinformatics projects in Inria, test the different applications, and contact project-teams. Eight project-teams participate in the development of this portal. Inria Bioscience Resources is developed in an Inria Technology Development Action (ADT).

⁵<http://altarica.labri.fr>

⁶<http://biorica.gforge.inria.fr>

⁷<http://www.genolevures.org/>

MANAO Team

4. Software and Platforms

4.1. EIGEN

Participants: G. Guennebaud, D. Nuytsa

Keywords : Linear algebra

Efficient numerical computation is central to many computer science domains. In particular, in computer graphics, space transformations and local regressions involve dense linear algebra, data interpolation and differential equations require sparse linear algebra, while more advanced problems involve non-linear optimization or spectral analysis. On the one hand, solutions such as MatLab are limited to prototyping. On the other hand, optimized libraries coming from the HPC (high performance computing) world are often tedious to use and more adapted for very large problems running on clusters. Moreover, all these solutions are very slow at handling very small but numerous problems which often arise in computer graphics, vision, or robotics. As a result, researchers of these domains used to waste a lot of time at either implementing their own half cooked solution, or dealing with dozens of complex to use libraries.

The objective of Eigen is to fill this gap by proposing an easy to use, efficient, and versatile C++ mathematical template library for linear algebra and related algorithms. In particular it provides fixed and dynamic size matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), sparse matrices with iterative and direct solvers, some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), some non-linear solvers, automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE, AltiVec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform global expression optimizations, and to remove unnecessary temporary objects.

Eigen is already a well established library with about 30k unique visitors of the website per month. Eigen is co-developed and maintained with a couple of other researchers and occasional contributors spread over the world. Its development started in 2008, and the last release is the 3.2 version in July 2013. Eigen has been supported by Inria through an ADT started in January 2012, and that ended in September 2013. This year, Eigen received the **“high-quality software in geometry processing award”** from the Symposium on Geometry Processing 2013 which was held in Genova, Pisa.

Facts:

- Web: <http://eigen.tuxfamily.org/>
- License: MPLv2

4.2. PatateLib

Participants: N. Mellado, G. Ciaudo, G. Guennebaud, P. Barla

Keywords : multi-scale analysis, material appearance, vector graphics, expressive rendering, 2D animation

Patate is a header only C++/CUDA library for graphics applications released under the MPL license.

It provides a collection of Computer Graphics techniques that incorporate the latest innovations from Inria research teams working in the field. It strives for efficiency and ease-of-use by focusing on low-level core operators and key algorithms, organised in modules that each tackle a specific set of issues. The central goal of the library is to drastically reduce the time and efforts required to turn a research paper into a ready-to-use solution, for both commercial and academic purposes.

Each module is initially developed by a few persons, usually those who have authored the corresponding research papers. An engineer, Gautier Ciaudo, has been recruited via the ADT program to perform unit tests, bug tracking, and make examples. Our first module provides efficient methods for the fitting and analysis of point-clouds in arbitrary dimensions. It may be used for varied purposes such as curvature computation, surface reconstruction, scale-space analysis, image processing, and sketch vectorization. More modules will be developed in 2014.

Facts:

- Web: <http://patate.gforge.inria.fr/html/index.html>
- License: MPLv2

4.3. PFSTools

Participant: I. Ihrke

Keywords : high dynamic range image processing, merging, calibration and tone-mapping

The `pfstools` package is a set of command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. All programs in the package exchange data using a simple generic high dynamic range image format, `pfs`, and they use unix pipes to pass data between programs and to construct complex image processing operations.

`pfstools` come with a library for reading and writing `pfs` files. The library can be used for writing custom applications that can integrate with the existing `pfstools` programs. It offer also a good integration with high-level mathematical programming languages, such as MATLAB or GNU Octave. `pfstools` can be used as the extension of MATLAB or Octave for reading and writing HDR images or simply to store effectively large matrices. The `pfstools` package is an attempt to integrate the existing high dynamic range image formats by providing a simple data format that can be used to exchange data between applications. It is accompanied by the `pfscalibration` and `pfstmo` packages.

The `pfscalibration` package provides an algorithm for the photometric calibration of cameras and for the recovery of high dynamic range (HDR) images from the set of low dynamic range (LDR) exposures. Maintenance of the `pfscalibration` package is done by Ivo Ihrke since January 2011. A major update to make the software compatible with current digital SLR cameras and their raw file formats, especially for measurement purposes, has been performed. A new set of MATLAB scripts has been developed for improved calibration performance. It is intended to merge these new procedures into the existing software.

The `pfstmo` package contains the implementation of seven state-of-the-art tone mapping operators suitable for convenient processing of both static images and animations.

The software received wider interest of the Open Source community and third party contributors prepared installation packages which are included in several Linux distributions including Debian, Fedora and Suse.

Facts:

- Web: <http://pfstools.sourceforge.net/>
- License: GPL

MC2 Project-Team

5. Software and Platforms

5.1. eLYSe

Participant: Olivier Saut.

eLYse is a numerical platform used for our computations in Biology (tumor growth), micro-fluidics and complex Newtonian fluid flows. The platform is divided in two libraries : one is devoted to the modelling equations and the other one includes the numerical solvers. For example, we are able to treat (in 2D and 3D) transport equations, diffusion equations, Navier-Stokes equations, Maxwell system and the interaction fluid-structure by level-set and penalization methods. The solvers are based on finite volume methods on cartesian grids and allow parallel computations. See also the web page <http://www.math.u-bordeaux1.fr/~osaut/pages/eLYSe.html>.

- Version: 0.7
- ACM: ACM J.2 J.3 G.1.8 G.1.10
- AMS: AMS65Z05 35Q92
- Keywords: Modélization and numerical simulations, Finite volume methods, Level Set approach, Penalization method
- APP: En cours
- Type of human computer interaction: console
- OS/Middleware: Platform developped on Mac OS X architecture.
- Required library or software: Petsc (<http://www.mcs.anl.gov/petsc/petsc-as/>) Vtk (<http://www.vtk.org/>) Blitz++ (<http://c2.com/cgi/wiki?BlitzPlusPlus>) (optionnel) Boost (<http://www.boost.org/>)
- Programming language: C++
- Documentation: doxygen.

5.2. Kesaco

Participant: Olivier Saut.

Kesaco is a set of libraries and programs aiming at applications of mathematical modeling in clinical oncology. It features:

- A library of specialized mathematical model describing the growth of different types of cancers (secondary tumors in the lung, gliomas).
- A set of programs useful to validate mathematical models (compute the various behavior they can produce) and to build databases of numerical simulations.
- Segmentation and registration routines to use medical images directly in our numerical codes.
- Calibration methods to recover the parameters of the models using sequences of medical images. Three techniques are implemented (a genetic algorithm, a technique based on reduced order models, a sensitivity technique).

All these routines are adapted to run on a MP architecture. The webpage may be found at <http://www.math.u-bordeaux1.fr/~osaut/pages/kesaco.html>.

- Version: 0.2
- Keywords: Modélization and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: eLYSe, Insight Toolkit (<http://www.itk.org>)
- Programming language: C++
- Documentation: doxygen.

5.3. NaSCar

Participant: Michel Bergmann [correspondant].

This code is devoted to solve 3D-flows in around moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam's type. The code is written in C language and use Petsc (<http://www.mcs.anl.gov/petsc/petsc-as/>) library for the resolution of large linear systems in parallel.

NaSCar can be used to simulate both hydrodynamic bio-locomation as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Version: 1
- Keywords: numerical analyse, fluid mechanics, langage C, PETSc
- Software benefit : simulate a flow around a deformable obstacle, moving into a fluid.
- APP: En cours
- Patent: non
- Type of human computer interaction: human for the moment
- OS/Middelware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progress

5.4. S-MPI-2D-3D

Participants: Charles-Henri Bruneau [correspondant], Khodor Khadra.

The software NS-MPI-2D-3D is a numerical platform devoted to the computation of the incompressible flow around bodies in two or three dimensions modelled by Stokes, Navier-Stokes or Oldroyd-B equations. It is based on finite differences or finite volumes approximations on cartesian grid using the volume penalization method to handle the obstacles. The resolution is achieved by means of the multigrid method. Dirichlet, periodic or artificial boundary conditions are implemented to solve various problems in closed or open domains.

- Version: 3
- Keywords: Numerical simulation of incompressible flows,
- Type of human computer interaction: console
- OS/Middelware: unix, linux, Mac OS X item Programming language: Fortran 95 and MPI
- Documentation: included

5.5. Other MC2 codes

- Penalization techniques on cartesian grids to solve incompressible Navier-Stokes equations
 - **Vortex**: sequential, Vortex In-Cell (VIC) scheme : hybrid vortex methods based on the combination of Lagrangian mesh-free schemes and Eulerian grid based schemes on the same flow region.
 - Unstructured body fitted meshes
 - **Richards** : 2D Unstructured finite element code, implicit solver, sequential, to solve the transport-diffusion equations through a porous media including tidal forcing and mechanisms of diagenesis.
 - development inside **FluidBox** software in collaboration with **BACCHUS**. 2D-3D unstructured meshes, Stabilized Finite Elements method (SUPG), RANS turbulence model, parallel: Domain Decomposition and MPI.
- Immersed boundary techniques for:
 - **Compressible flows** : 2D-3D finite volume scheme for compressible Euler equations with solid obstacles on cartesian grids. 3D code parallelized with MPI
 - **Elliptic problems** : 2 2D-3D finite difference scheme for elliptic interface problems, parallelized with PETSc
 - Electroporabilization: 2D finite difference scheme, parallelized with PETSc to simulate the electroporabilization of biological cells

MNEMOSYNE Team

5. Software and Platforms

5.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [48], [45], [50] have already made us define a library (DANA [44]), closely related to both the notion of artificial neural networks and cellular automata. From a conceptual point of view, the computational paradigm supporting the library is grounded on the notion of a unit that is essentially a (vector of) potential that can vary along time under the influence of other units and learning. Those units can be organized into layers, maps and networks.

We also gather in the middleware EnaS (that stands for *Event Neural Assembly Simulation*; cf. <http://gforge.inria.fr/projects/enas>) our numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". In 2013, we have also integrated in this C/C++ middleware early-vision perception routines developed within the scope of the KEOpS project.

We will also have to interact with the High Performance Computing (HPC) community, since having large scale simulations at that mesoscopic level is an important challenge in our systemic view of computational neuroscience. Our approach implies to emulate the dynamics of thousands, or even millions, of integrated computational units, each of them playing the role of a whole elementary neural circuit (e.g. the microcolumn for the cortex). Mesoscopic models are considered in such an integrative approach, in order to exhibit global dynamical effect that would be hardly reachable by compartment models involving membrane equations or even spiking neuron networks.

The vast majority of high performance computing softwares for computational neuroscience addresses sub-neural or neural models [34], but coarser grained population models are also demanding for large scale simulations, with fully distributed computations, without global memory or time reference, as it is specified in (cf. § 3.2).

5.2. Dana

Participant: Nicolas Rougier.

DANA [44] is a python framework (<http://dana.loria.fr>) whose computational paradigm is grounded on the notion of a unit that is essentially a set of time dependent values varying under the influence of other units via adaptive weighted connections. The evolutions of a unit's value are defined by a set of differential equations expressed in standard mathematical notation which greatly ease their definition. The units are organized into groups that form a model. Each unit can be connected to any other unit (including itself) using a weighted connection. The DANA framework offers a set of core objects needed to design and run such models. The modeler only has to define the equations of a unit as well as the equations governing the training of the connections. The simulation is completely transparent to the modeler and is handled by DANA. This allows DANA to be used for a wide range of numerical and distributed models as long as they fit the proposed framework (e.g. cellular automata, reaction-diffusion system, decentralized neural networks, recurrent neural networks, kernel-based image processing, etc.).

5.3. ENAS: Event Neural Assembly Simulation

Participants: Frédéric Alexandre, Nicolas Rougier, Thierry Viéville.

EnaS (that stands for “Event Neural Assembly Simulation”) is a middleware implementing our last numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". The recent achievements include (in collaboration with the Neuromathcomp EPI): spike trains statistical analysis via Gibbs distributions, spiking network programming for exact event's sequence restitution, discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods. On the mnemosyne side, this middleware includes since 2013 functional simulations of the non-standard perceptive behavior of the retina (detection of visual events) based on intensity and local spatial intensity cues, while in 2014 we are going to extend these new developments to motion cues. We have also made a strong effort in terms of interoperability on our side, since main functions are now usable from other languages (especially the python wrapper, actually in use), while we still maintain the interoperability with . This development is a complement of what has been developed on the Neuromathcomp side where colleagues have invested in a GUI for their routines.

5.4. Virtual Enaction

Participants: Frédéric Alexandre, André Garenne, Nicolas Rougier, Thierry Viéville.

The computational models studied in this project have applications that extend far beyond what is possible to experiment yet in human or non-human primate subjects. Real robotics experimentations are also impaired by rather heavy technological constraints; for instance, it is not easy to dismantle a given embedded system in the course of emerging ideas. The only versatile environment in which such complex behaviors can be studied both globally and at the level of details of the available modeling is a virtual environment, as in video games. Such a system can be implemented as “brainy-bot” (a programmed player based on our knowledge of the brain architecture) which goal is to survive in a complete manipulable environment.

In order to attain this rather ambitious objective we are going to both (i) deploy an existing open-source video game middleware in order to be able to shape the survival situation to be studied and (ii) revisit the existing models in order to be able to integrate them as an effective brainy-bot. It will consist of a platform associated to a scenario that would be the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it would be easy to integrate an artificial agent with sensory inputs (visual, touch and smell), emotional and somatosensory cues (hunger, thirst, fear, ..) and motor outputs (movement, gesture, ..) connected to a "brain" whose architecture will correspond to the major anatomical regions involved in the issues of learning and action selection (cortex areas detailed here, basal ganglia, hippocampus, and areas dedicated to sensorimotor processes). The internal game clock will be slowed down enough to be able to run non trivial brainy-bot implementations. This platform has already being used by two students of the team and is now a new deliverable of the KEOpS project.

PHOENIX Project-Team

5. Software and Platforms

5.1. DiaSuite: a Development Environment for Sense/Compute/Control

Applications

Participants: Charles Consel [correspondent], Damien Martin-Guillerez, Milan Kabac, Paul Van Der Walt, Camille Manano, Adrien Carteron, Alexandre Spriet, Emilie Balland.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DIASUITE is a suite of tools covering the development life-cycle of a pervasive computing application:

- *Defining an application area.* First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the DIASPEC language.
- *Designing an application.* Given a taxonomy, the architect can design and structure applications. To do so, the DIASPEC language provides an application design layer [31]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [27]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.
- *Implementing an application.* We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the DIAGEN compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.
- *Testing an application.* DIAGEN generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the DIASIM tool, without requiring any code modification. DIASIM provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.
- *Deploying a system.* Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 2 .

See also the web page <http://diasuite.inria.fr>.

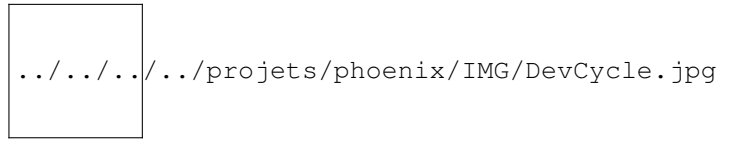


Figure 2. DIASUITE Development Cycle

5.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DIASUITE development environment is the domain specific language called DIASPEC and its compiler DIAGEN:

- DIASPEC is composed of two layers:
 - The *Taxonomy Layer* allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.
 - The *Architecture Layer* is based on an architectural pattern commonly used in the pervasive computing domain [27]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (e.g., varying light management implementations in a controller component to optimize energy consumption).
- DIAGEN is the DIASPEC compiler that performs both static and runtime verifications over DIASPEC declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DIAGEN supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

5.1.2. DiaSim: a Parametrized Simulator for Pervasive Computing Applications

Pervasive computing applications involve both software and integration concerns. This situation is problematic for testing pervasive computing applications because it requires acquiring, testing and interfacing a variety of software and hardware entities. This process can rapidly become costly and time-consuming when the target environment involves many entities.

To ease the testing of pervasive applications, we are developing a simulator for pervasive computing applications: DIASIM. To cope with widely heterogeneous entities, DIASIM is parameterized with respect to a DIASPEC specification describing a target pervasive computing environment. This description is used to generate with DIAGEN both a programming framework to develop the simulation logic and an emulation layer to execute applications. Furthermore, a simulation renderer is coupled to DIASIM to allow a simulated pervasive system to be visually monitored and debugged. The simulation renderer is illustrated in Figure 3.



Figure 3. A screenshot of the DIASIM simulator

5.2. DiaSuiteBox: an Open Orchestration Platform

Participants: Julien Bruneau [correspondent], Damien Martin-Guillerez, Quentin Enard, Charles Consel.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIASUITE project.

DiaSuiteBox platform architecture The DiaSuiteBox platform can be embedded in a small plug-computer or deployed in the cloud. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured the behavior of its applications are innocuous and correct with respect to the provided information. Finally, DiaSuiteBox provides an extensible software architecture. This allows the easily connect new device technologies to the platform. For example, the support for new wireless communication technologies such as Zigbee, Z-Wave or Sigfox can be easily added to the DiaSuiteBox platform.

More details can be found on the web page <http://diasuitebox.inria.fr>.

The iQSpot startup uses DiaSuiteBox as a software platform to ease the management of Smart Buildings. In this project, the DiaSuiteBox platform is first used to host building management functionalities such as lighting management, heating/ventilating/air conditioning management, energy efficiency monitoring. It is also used to host software drivers that allow the building management functionalities to interact with the connected devices deployed in buildings. These devices can use wired communication technologies such LonWorks, BACNet or KNX, as well as wireless communication technologies such as Z-Wave or Zigbee.

5.3. School+ Apps: Assistive tablet applications for school Inclusion

Participants: Charles Consel [correspondent], Charles Fage, Damien Martin-Guillerez, Camille Manano, H el ene Sauz eon.

College+ is a package of 7 applications. Three applications are assistive applications, guiding the child doing specific tasks. Three others are training applications made as serious games, addressing specific skills. The last application is a meta-application, comprising a link to the three training applications, with an access to statistics of their usage. For each application, data are separated from the design, meaning that every element of each application (pictures, texts, settings, etc.) can be changed at any time. Each application records a log file every interactions performed by the child.

Assistive applications :

- *Routines application.* This application shows a list of tasks, with a short description. After clicking the starting button, a specific slideshow is shown ; it decomposes a task into steps. For each step, a text and a picture can be displayed. Thumbnail of previous and next steps are also displayed. This application guides the child through classroom situations: entering classroom, taking school materials out of a backpack, writing notes, handling agenda, leaving the classroom.
- *Communication application.* With the same design, the assistance provided by this application targets to communicating situations inside the classroom. The application covers four scenarios addressing two interaction situations (initiating and answering the interaction) and two types of interlocutors (professor and classmate). For each scenario, different slideshows guide the child, depending on the goal of the interaction.
- *Emotion Regulation application.* This application aims to assist the child to self-regulate his/her emotions. Four simplified emoticons are proposed to the child to choose from: anger, sadness, joy and fear. Then, (s)he selects a level of intensity via a thermometer with a scale from 1 to 4. In response, the application delivers different multimedia contents according to the level selected to help the child regulate his/her emotions. Typically, a text (breathing instructions) are shown at level 1, pictures at level 2, a video at level 3 and another text at level 4.



Figure 4. Assistive applications



Figure 5. Assistive applications

Training applications: These three applications are serious games with increasing levels of difficulties, reachable after a ratio of good answers has been attained.

- *Emotion Recognition application: pictures.* In this application, the child is instructed to identify a specific emotion among 4 pictures showing different people exhibiting an emotion. Seven emotions are involved in this application: joy, sadness, fear, anger, surprise, disgust and neutral. The emotion to be recognized is displayed together with its simplified emoticon. The type of pictures changes with the difficulty level: level 1 contains pictures of unfamiliar people and level 2 contains pictures of friends and relatives of the child.
- *Emotion Recognition application: videos.* In this application, the child is presented with a fragment of an animated cartoon. At some point, the video stops and the child is asked to identify the emotion of the character. Four emotions are involved in this application: joy, sadness, fear and anger. Videos are slowed down, with a speed percentage that can be changed at each level. Videos change with difficulty level: level 1 contains videos of a very basic cartoon (only one cartoon character drawn by basic form un-textured), level 2 contains a video of more sophisticated cartoons and level 3 contains movies with actors.
- *Attention Training.* In this application, the child is presented a picture of a face and asked to make eye contact with it. Second, a symbol appears briefly in the eyes of the character. Third, the child is asked to identify the symbol shown in the previously displayed picture, to make sure he kept eye contact. The speed at which the symbol appears and disappears is changed according to the difficulty level. Types of pictures also change with the level : level 1 contains pictures of faces and level 2 contains pictures of classroom situations.

POTIOC Team

5. Software and Platforms

5.1. OpenViBE

Participants: Fabien Lotte [local correspondent], Alison Cellard [engineer].

As part of our research work on BCI, we contribute to the development of the OpenViBE² software, which is an open source platform dedicated to the design, evaluation and use of BCI for real and virtual applications. OpenViBE development is led by Inria, and Potioc is one of the Inria team contributing to its evolution. Moreover, Potioc is involved in the Inria ADT (Technological Development Action) OpenViBE-NT that is dedicated to the development of OpenViBE together with 3 other Inria teams (Hybrid, Athena, Neurosys).

5.2. Drile

Participant: Florent Berthaut.

As part of the research on Virtual Reality for Musical Performance, notably the Drile system, various software pieces are being developed and made available to the community. These software pieces are the following:

- Pure-Data external to access data from the Virtual Reality Peripheral Network : <https://github.com/scrime/vrpd>
- Drile: <http://hitmuri.net/index.php/Research/Drile>

5.3. 3DCityTestbed

Participants: Jacek Jankowski, Thomas Hulin.

As part of the research on the "Villes transparentes" project in collaboration with Mappy (Solocal group) and Vectuel - VirtuelCity, we develop a software platform dedicated to the design and evaluation of innovative interaction techniques for the navigation in urban 3D environments. This code is not publicly available yet.

5.4. PapARt

Participant: Jérémy Laviolle [Main developer].

As part of his thesis work, Jérémy Laviolle has developed a software suite for PapARt : Paper Augmented Reality Toolkit. This suite enables the calibration of depth cameras such as the Kinect with a planar surface and with a videoprojection. It also enables the detection of finger touch on the planar surface. This system can be extended to 3D objects. Many external devices are compatible, such as pen tablets (Wacom) and LEAP Motion. The code runs on Linux, Mac OS and Windows, and is made for Processing³. This code is not publicly available yet.

²<http://openvibe.inria.fr>

³<http://www.processing.org>

REALOPT Project-Team

5. Software and Platforms

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Romain Leguay [Software Engineer], Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of a Dantzig-Wolfe reformulation technique. The reformulated problem is solved using a branch-and-price (column generation) algorithm. This software platform, made of C++ classes, offers a “*black-box*” implementation that does not require user input and is not application specific. The features are

(i) the automation of the Dantzig-Wolfe reformulation process (the user defines a mixed integer programming problem in a pseudo modeling language, defining variables and constraints, identifying subproblems. He can provide subproblem solvers if available, but he does not need to explicitly define the reformulation, the explicit form of the columns, their reduced cost, or the Lagrangian bounds.

(ii) a default column generation procedure with standard initialization and stabilization [1], [59] [89] [88] [27] and

(iii) a default branching scheme that is generic to all applications [7],

(iv) default primal heuristics specially developed for use in a decomposition framework [64], [79], [90].

The prototype software was/is used as background solver in our application studies and local PhD thesis. It also serves as the framework for our comparative study in a Inria associated team project and our transfert projects (the prototype enables us to be very responsive in our industrial contact).

See also the web page <https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod>.

RUNTIME Project-Team

5. Software and Platforms

5.1. Common Communication Interface

Participant: Brice Goglin.

- The *Common Communication Interface* aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the *Oak Ridge National Laboratory* and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- <http://www.cci-forum.org>

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 30 000 lines of C.
- <http://runtime.bordeaux.inria.fr/hwloc/>

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (HWLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- HWLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement.
- HWLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C.
- <http://netloc.org>

5.4. KNem

Participant: Brice Goglin.

- KNEM (*Kernel Nemesis*) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 8 000 lines of C. Its main contributor is Brice GOGLIN.
- <http://runtime.bordeaux.inria.fr/knem/>

5.5. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [42] or the recent iWarp [41] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- <http://open-mx.org/>

5.6. StarPU

Participants: Olivier Aumage, Andra Hugo, Nathalie Furmento, Raymond Namyst, Marc Sergent, Samuel Thibault, Pierre-André Wacrenier.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi processors.
- STARPU offers a unified offloadable task abstraction named codelet. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).

- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- <http://runtime.bordeaux.inria.fr/StarPU/>

5.7. NewMadeleine

Participant: Alexandre Denis.

- NEWMADELEINE is communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- NEWMADELEINE targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization *strategies*, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- The reference software development branch of the NEWMADELEINE software consists in 90 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- <http://runtime.bordeaux.inria.fr/newmadeleine/>

5.8. PadicoTM

Participant: Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in *NEWMARIELEINE*. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60 000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- <http://runtime.bordeaux.inria.fr/PadicoTM/>

5.9. MAQAO

Participants: Denis Barthou, Olivier Aumage, Tamara Meunier.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes for Intel x86 and ARM architectures. For x86 architecture, it can insert probes for instrumentation directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.
- MAQAO has a static performance model for x86 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the European FP7 project “MontBlanc” and in the Samsung GRO project “Gepetto”.
- <http://www.maqao.org/>

5.10. QIRAL

Participants: Denis Barthou, Olivier Aumage.

- QIRAL is a high level language (expressed through LaTeX) that is used to describe Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developed in collaboration with other teams participating to the ANR PetaQCD project.

5.11. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size.
- Some core algorithms are parallel to speed-up the execution.
- TREEMATCH is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.
- TREEMATCH is available at: <http://treematch.gforge.inria.fr>.